



第4章 组合逻辑电路

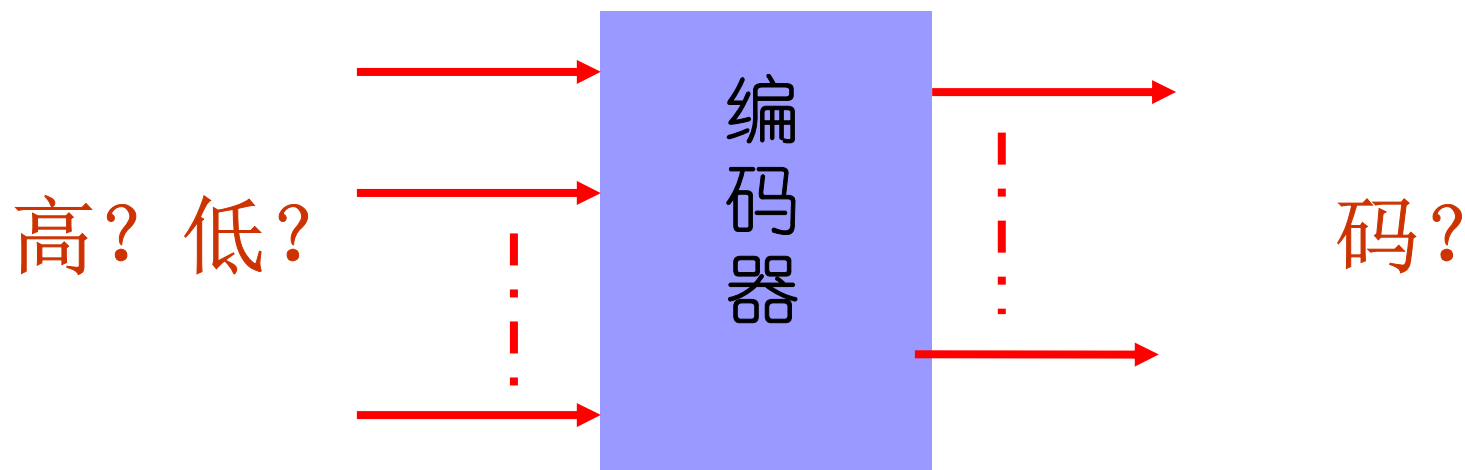
第2讲：常用组合逻辑电路

4.3 若干常用的组合逻辑电路

§ 4.3.1 编码器 (Encoder)

编码：用二进制代码来表示某一信息（文字、数字、符号）的过程。

实现编码操作的电路称为**编码器**。





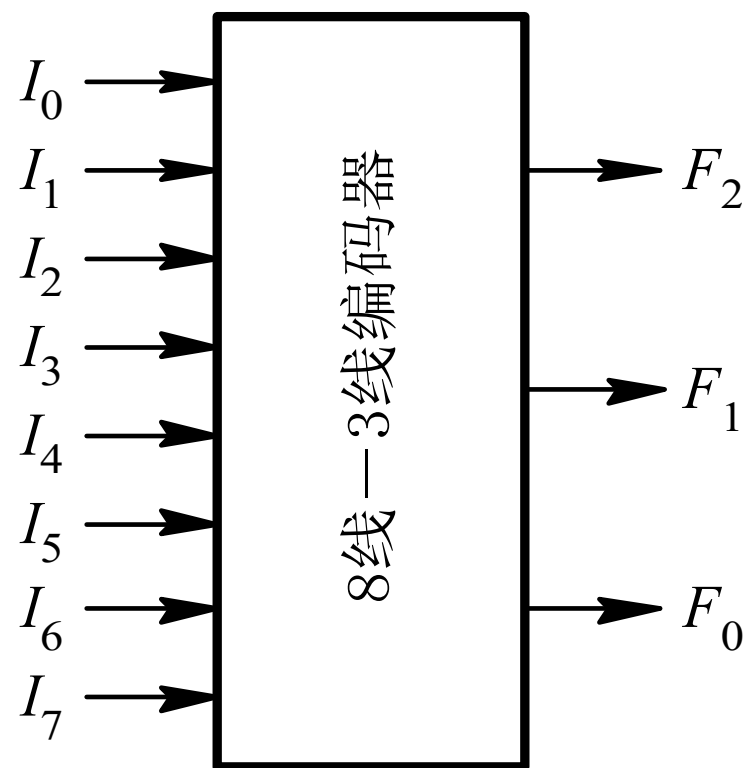
二进制编码器

用 n 位二进制代码对 $N=2^n$ 个一般信号进行编码的电路，叫做**二进制编码器**。例如 $n=3$ ，可以对8个一般信号进行编码。

特点：任何时刻只允许输入一个有效信号，不允许同时出现两个或两个以上的有效信号，因而其输入是一组有约束(互相排斥)的变量。

编码器的工作原理:

下图是三位二进制编码器的框图，它的输入是 $I_0 \sim I_7$ 8个高电平信号，输出是三位二进制代码 F_2 、 F_1 、 F_0 。为此，又把它叫做8线—3线编码器。输出与输入的对应关系如表所示。



三位二进制8线—3线编码器框图



三位二进制编码器的真值表

| 输 入 | | | | | | | | 输 出 | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| I_0 | I_1 | I_2 | I_3 | I_4 | I_5 | I_6 | I_7 | F_2 | F_1 | F_0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |



由真值表求最小项之和，可得出编码器的输出函数为

$$F_2 = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 I_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 I_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

$$F_1 = \bar{I}_0 \bar{I}_1 I_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 I_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

$$F_0 = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 I_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

| I_0 | I_1 | I_2 | I_3 | I_4 | I_5 | I_6 | I_7 | F_2 | F_1 | F_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |



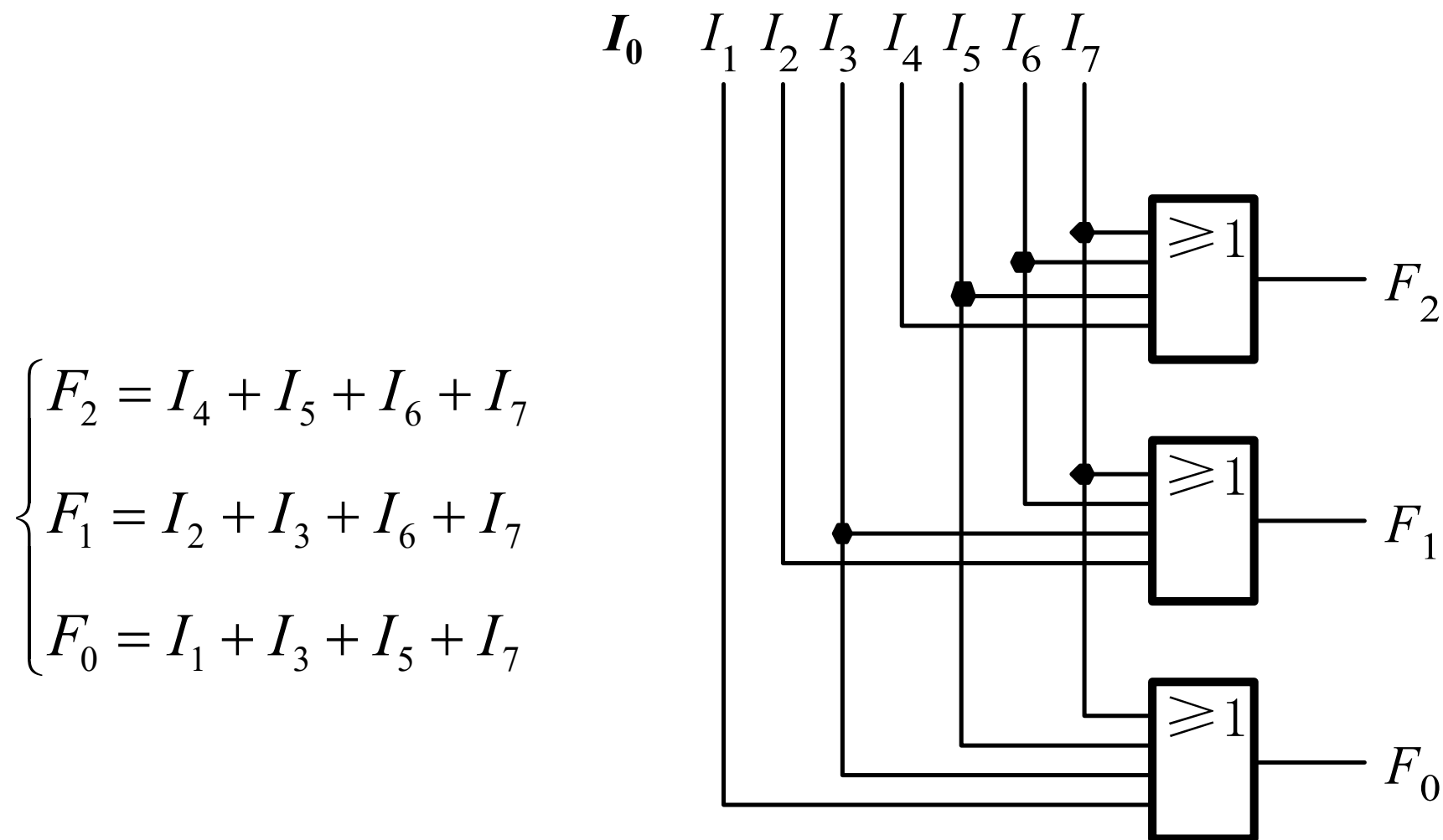
$$F_2 = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 I_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 I_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

$$F_1 = \bar{I}_0 \bar{I}_1 I_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 I_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

$$F_0 = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 I_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

因为任何时刻 $I_0 \sim I_7$ 当中仅有一个取值为1，利用这个约束条件将上式化简，得到

$$\begin{cases} F_2 = I_4 + I_5 + I_6 + I_7 \\ F_1 = I_2 + I_3 + I_6 + I_7 \\ F_0 = I_1 + I_3 + I_5 + I_7 \end{cases}$$



三位二进制编码器

的编码是隐含的，当 $I_1 \sim I_7$ 均为0时，电路的输出就是 I_0 的编码。



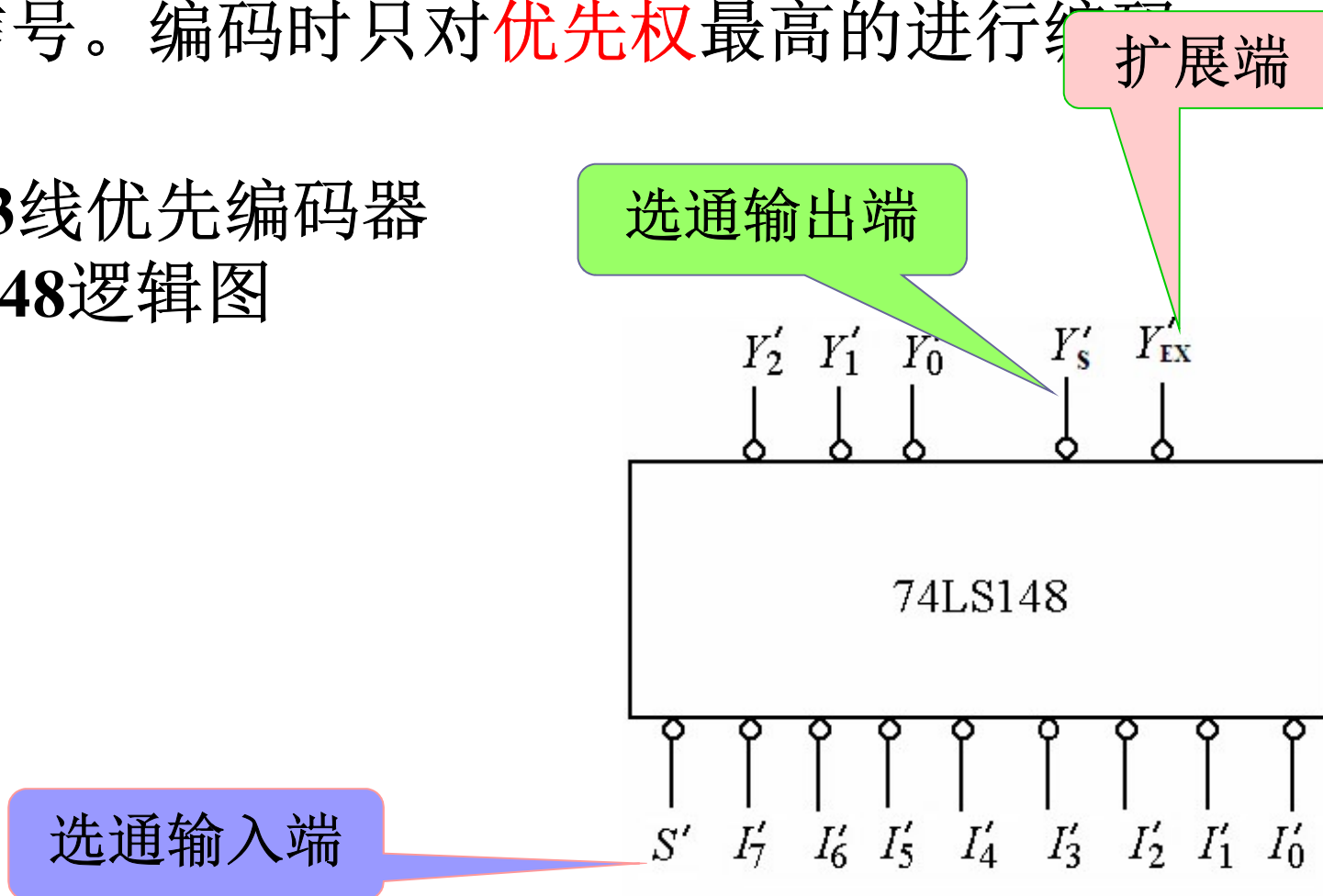
优先编码器 (Priority Encoder)

优先编码器常用于优先中断系统和键盘编码。与普通编码器不同，**优先编码器允许多个输入信号同时有效，但它只对其中优先级别最高的有效输入信号编码**，对级别较低的输入信号不予理会。

常用的 MSI 优先编码器有 10 线—4 线 (如 74LS147)、8 线—3 线 (如 74LS148)。

在优先编码器电路中，允许同时输入两个以上编码信号。编码时只对**优先权**最高的进行编码

8线—3线优先编码器 74LS148逻辑图



(b) 逻辑功能示意图

| 输 入 | | | | | | | | | 输 出 | | | |
|------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------------------|
| S' | I'_7 | I'_6 | I'_5 | I'_4 | I'_3 | I'_2 | I'_1 | I'_0 | Y'_2 | Y'_1 | Y'_0 | Y'_S Y'_{EX} |
| 1 | × | × | × | × | × | × | × | × | 1 | 1 | 1 | 1 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 1 |
| 0 | 0 | × | × | × | × | × | × | × | 0 | 0 | 0 | 1 0 |
| 0 | 1 | 0 | × | × | × | | | | | | 1 | 1 0 |
| 0 | 1 | 1 | 0 | × | × | | | | | | 0 | 1 0 |
| 0 | 1 | 1 | 1 | 0 | × | × | × | × | 0 | 1 | 1 | 1 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | × | × | × | 1 | 0 | | 1 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | × | | | | | 1 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | | 1 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 0 |

低电平表示“电路工作，但无编码输入”

低电平表示“电路工作，且有编码输入”

输入：逻辑0(低电平)有效

输出：逻辑0(低电平)有效

TI SN74LS148 8-Line-to-3-Line Encoder

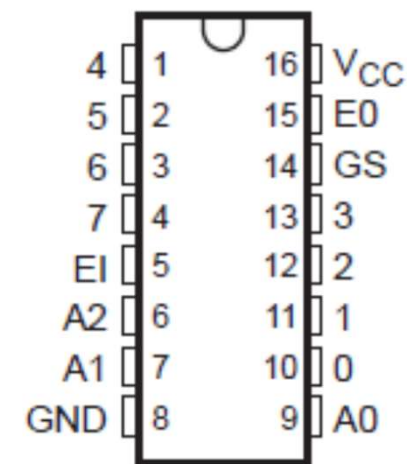
148, 'LS148

Encode 8 Data Lines to 3-Line
Binary(Octal)

Applications Include:

- n-Bit Encoding
- Code Converters and Generators

SN54148, SN54LS148 . . . J OR W PACKAGE
SN74148, SN74LS148 . . . D, N, OR NS PACKAGE
(TOP VIEW)



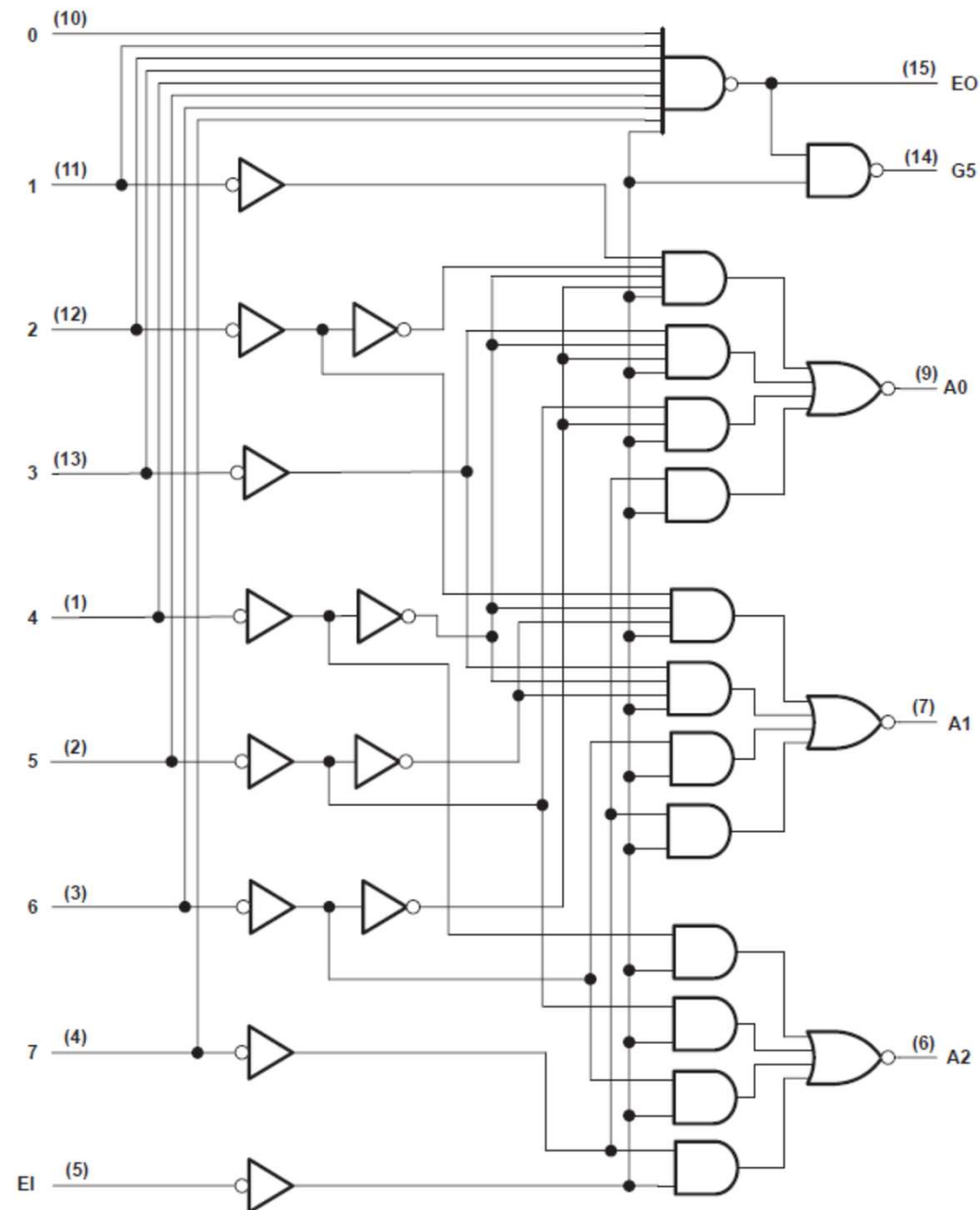


FUNCTION TABLE – '148, 'LS148

| INPUTS | | | | | | | | | OUTPUTS | | | | |
|--------|---|---|---|---|---|---|---|---|---------|----|----|----|----|
| EI | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | A2 | A1 | A0 | GS | EO |
| H | X | X | X | X | X | X | X | X | H | H | H | H | H |
| L | H | H | H | H | H | H | H | H | H | H | H | H | L |
| L | X | X | X | X | X | X | X | L | L | L | L | L | H |
| L | X | X | X | X | X | X | L | H | L | L | H | L | H |
| L | X | X | X | X | X | L | H | H | L | H | L | L | H |
| L | X | X | X | X | L | H | H | H | L | H | H | L | H |
| L | X | X | X | L | H | H | H | H | H | L | L | L | H |
| L | X | X | L | H | H | H | H | H | H | L | H | L | H |
| L | X | L | H | H | H | H | H | H | H | H | L | L | H |
| L | L | H | H | H | H | H | H | H | H | H | H | L | H |

H = high logic level, L = low logic level, X = irrelevant

'148, 'LS148 logic diagram (positive logic)

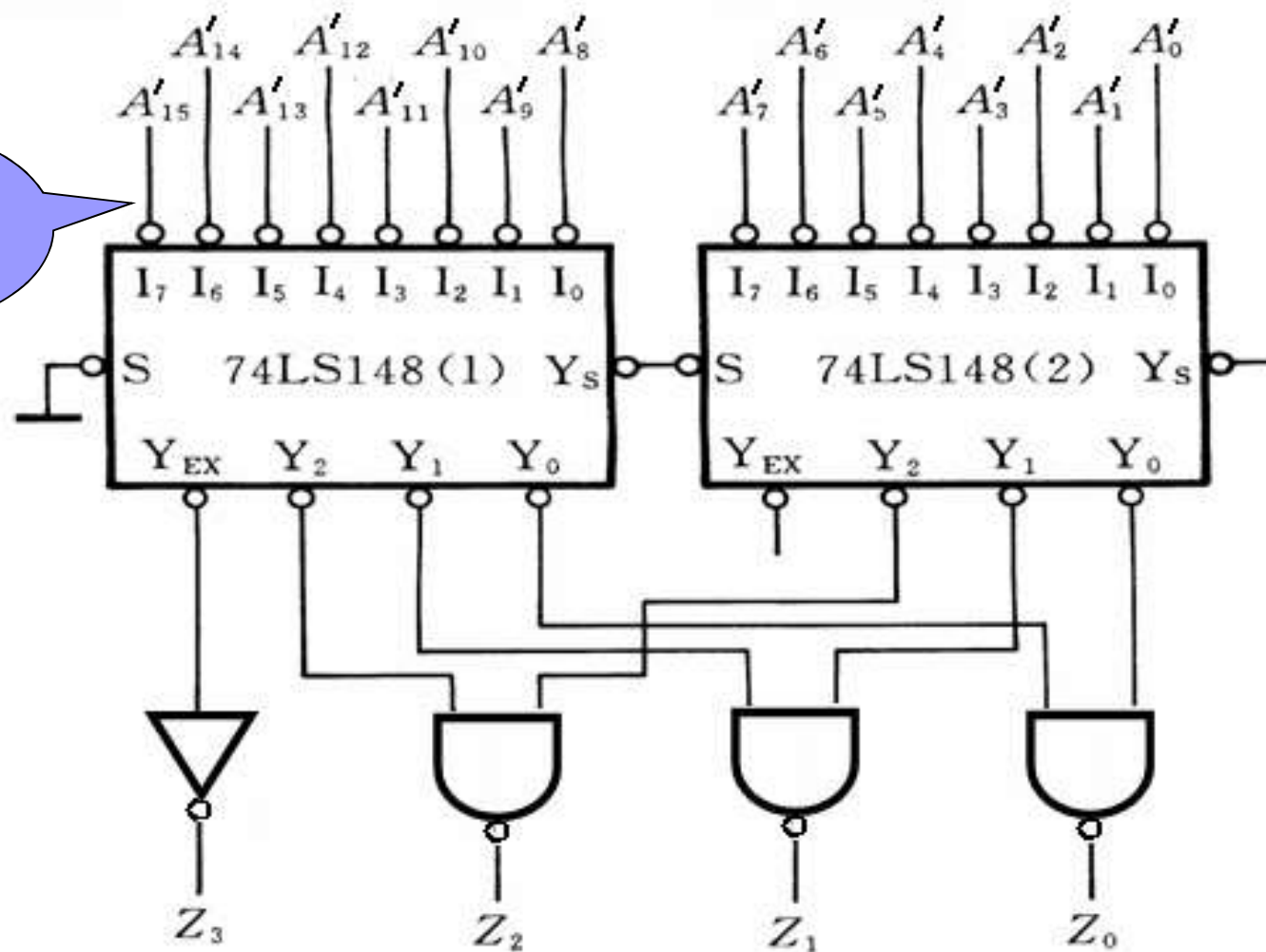


例：试用两片74LS148组成16线—4线优先编码器。

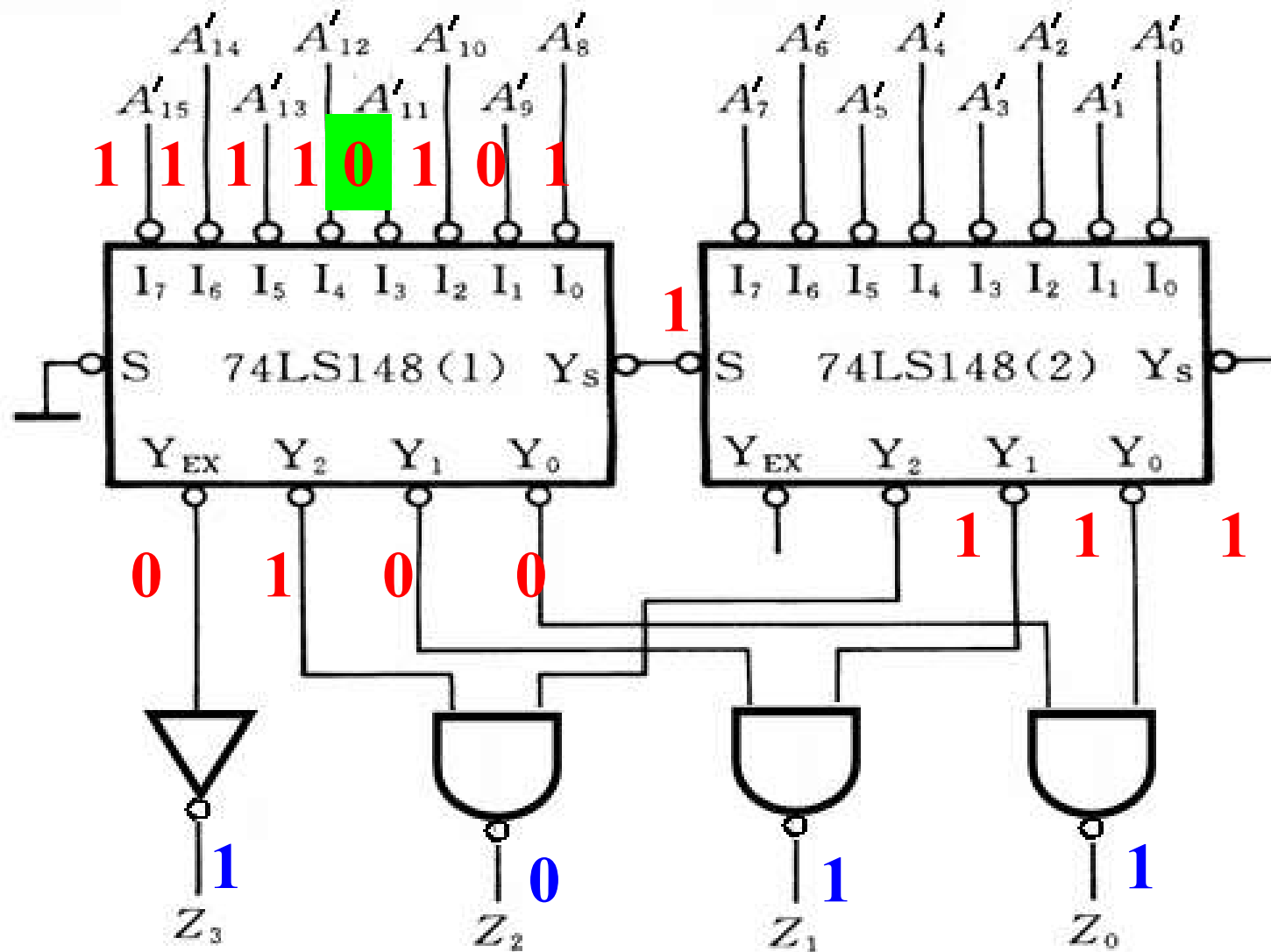
两片74LS148级联构成的16~4编码器真值简表

| $I_{15} \sim I_8$ | $I_7 \sim I_0$ | D | C | B | A |
|-------------------|----------------|-------|-----|-----|-----|
| 0 x x... | x...x | 0 | 0 | 0 | 0 |
| | | | | | |
| 1.....0 | x...x | 0 | 1 | 1 | 1 |
| 1.....1 | 0xx.. | 1 | 0 | 0 | 0 |
| 1.....1 | | | | | |
| 1.....1 | 1.....0 | 1 | 1 | 1 | 1 |

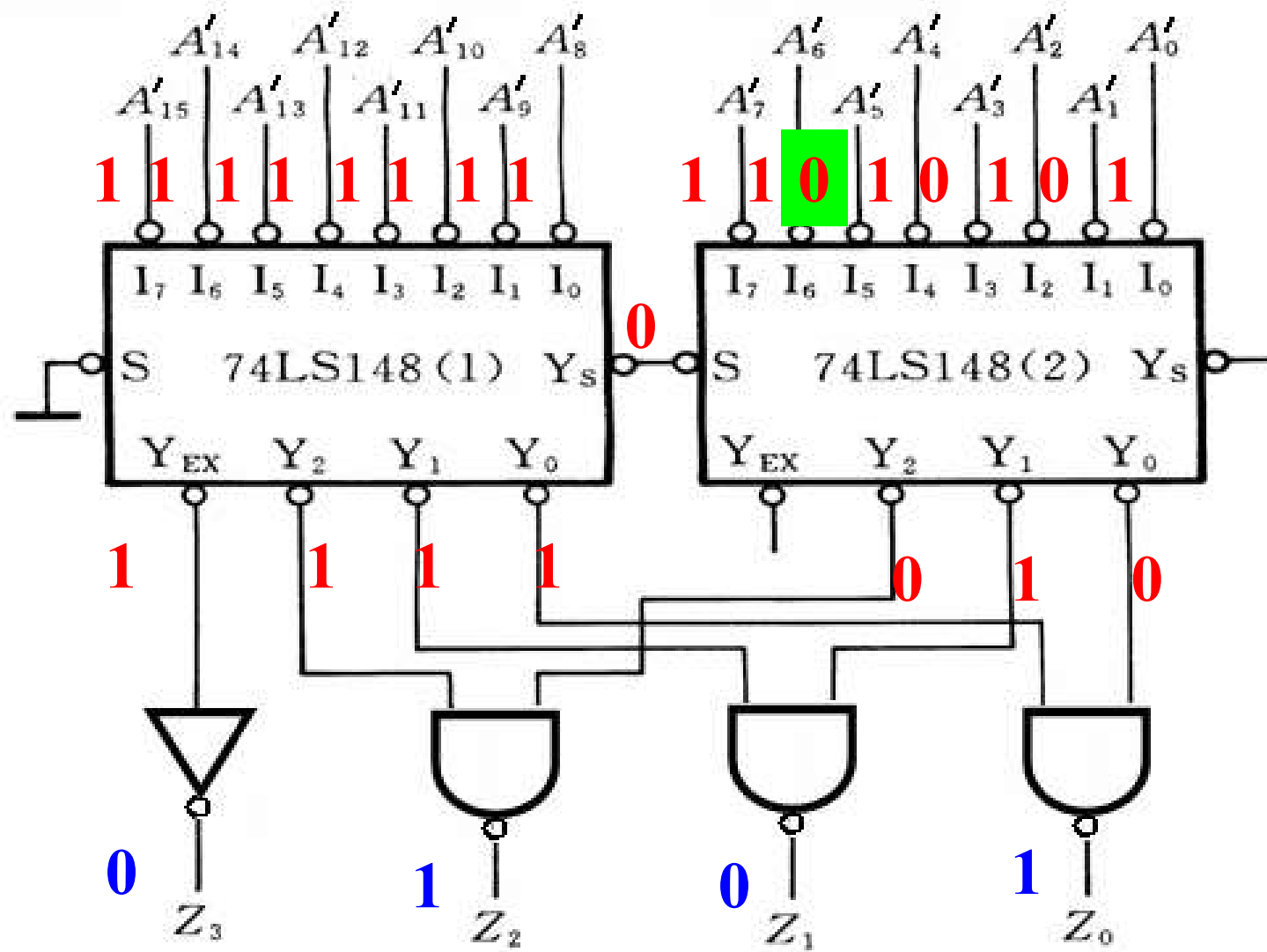
优先权
最高



$A'_{15} \sim A'_8$ 均无信号时，才允许对 $A'_7 \sim A'_0$ 输入信号编码。



(1)片处于编码状态,(2)片被封锁。

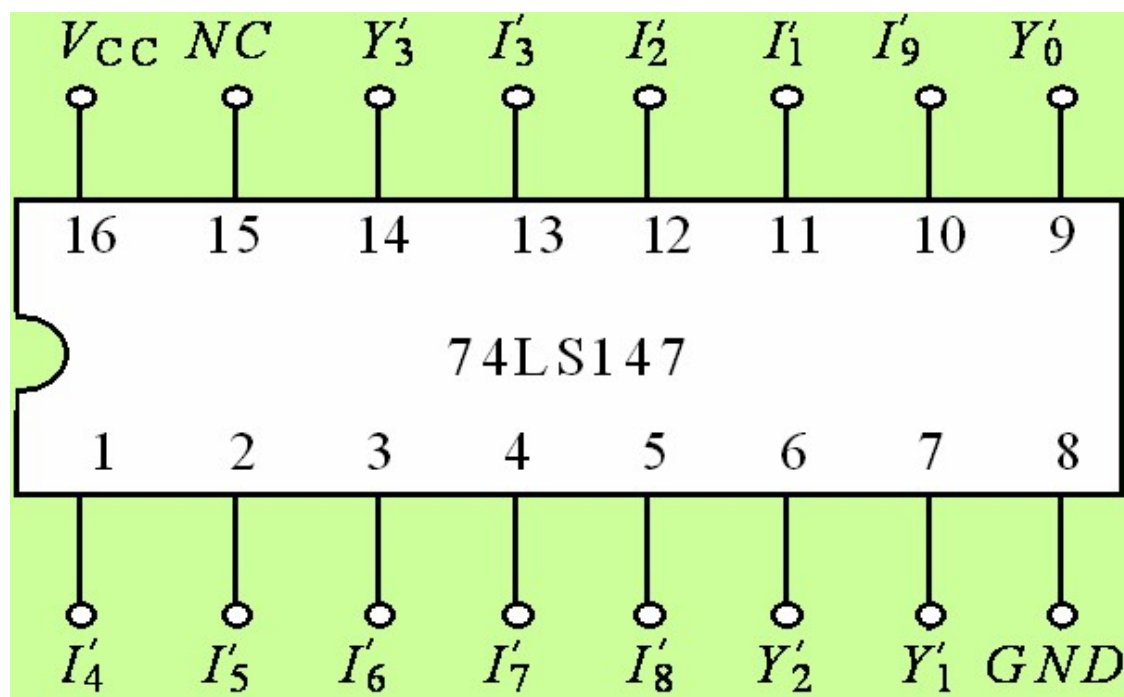


(2)片处于编码状态

二一十进制编码器 (Decimal-to-BCD Encoder)

输入端10个，输出端4个，也称10线—4线编码器。

集成10线-4线优先编码器



输入输出均低电平有效。

集成10线-4线优先编码器

FUNCTION TABLE – '147, 'LS147

| INPUTS | | | | | | | | | OUTPUTS | | | |
|--------|---|---|---|---|---|---|---|---|---------|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | D | C | B | A |
| H | H | H | H | H | H | H | H | H | H | H | H | H |
| X | X | X | X | X | X | X | X | L | L | H | H | L |
| X | X | X | X | X | X | X | L | H | L | H | H | H |
| X | X | X | X | X | X | L | H | H | H | L | L | L |
| X | X | X | X | X | L | H | H | H | H | L | L | H |
| X | X | X | X | L | H | H | H | H | H | L | H | L |
| X | X | X | L | H | H | H | H | H | H | L | H | H |
| X | X | L | H | H | H | H | H | H | H | H | L | L |
| X | L | H | H | H | H | H | H | H | H | H | L | H |
| L | H | H | H | H | H | H | H | H | H | H | H | L |

H = high logic level, L = low logic level, X = irrelevant



§ 4.3.2 译码器(Decoder)

译码：将二进制代码翻译成对应的输出信号的过程。译码是编码的逆过程。

实现译码操作的电路称为**译码器**。

常用的译码器有：二进制译码器、**BCD**一十进制译码器、显示译码器三类。



一、二进制译码器(Basic Binary Decoder)

输入端： n 输出端： 2^n

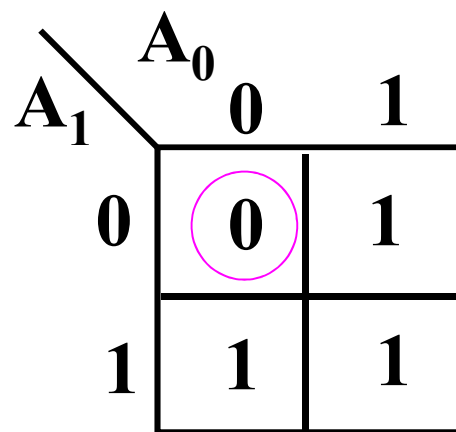
二进制译码器的输入端为 n 个，则输出端为 2^n 个，且对应于输入代码的每一种状态， 2^n 个输出中只有一个为1（或为0），其余全为0（或为1）。

2 线—4线译码器74LS139（输出低电平有效）

真值表

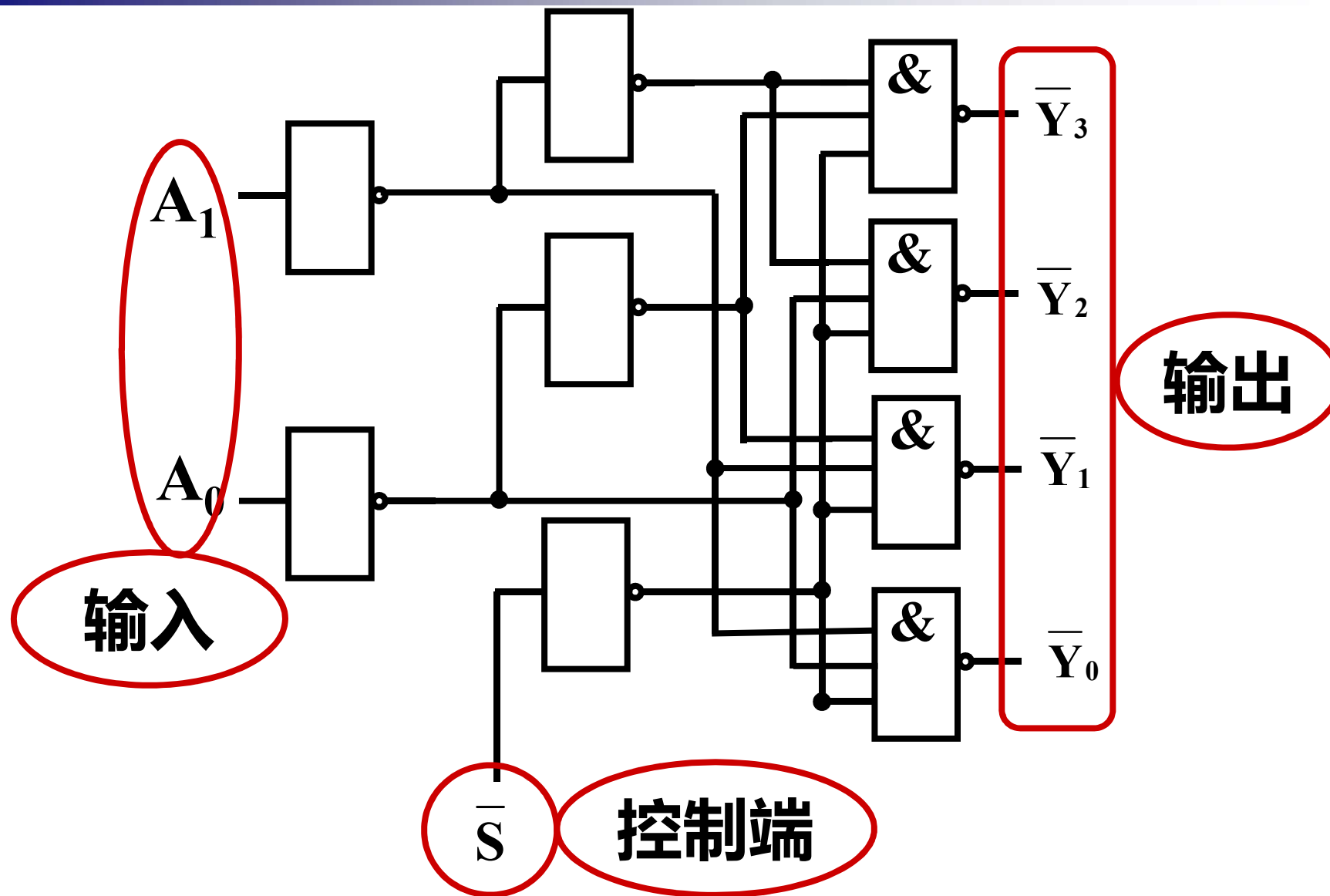
| A_1 | A_0 | Y_3' | Y_2' | Y_1' | Y_0' |
|-------|-------|--------|--------|--------|--------|
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |

画关于 Y_0' 的卡诺图



$$Y_0' = (A_1' \cdot A_0')' \quad Y_1' = (A_1' \cdot A_0)'$$

$$Y_2' = (A_1 \cdot A_0')' \quad Y_3' = (A_1 \cdot A_0)'$$



2-4译码器74LS139的内部线路

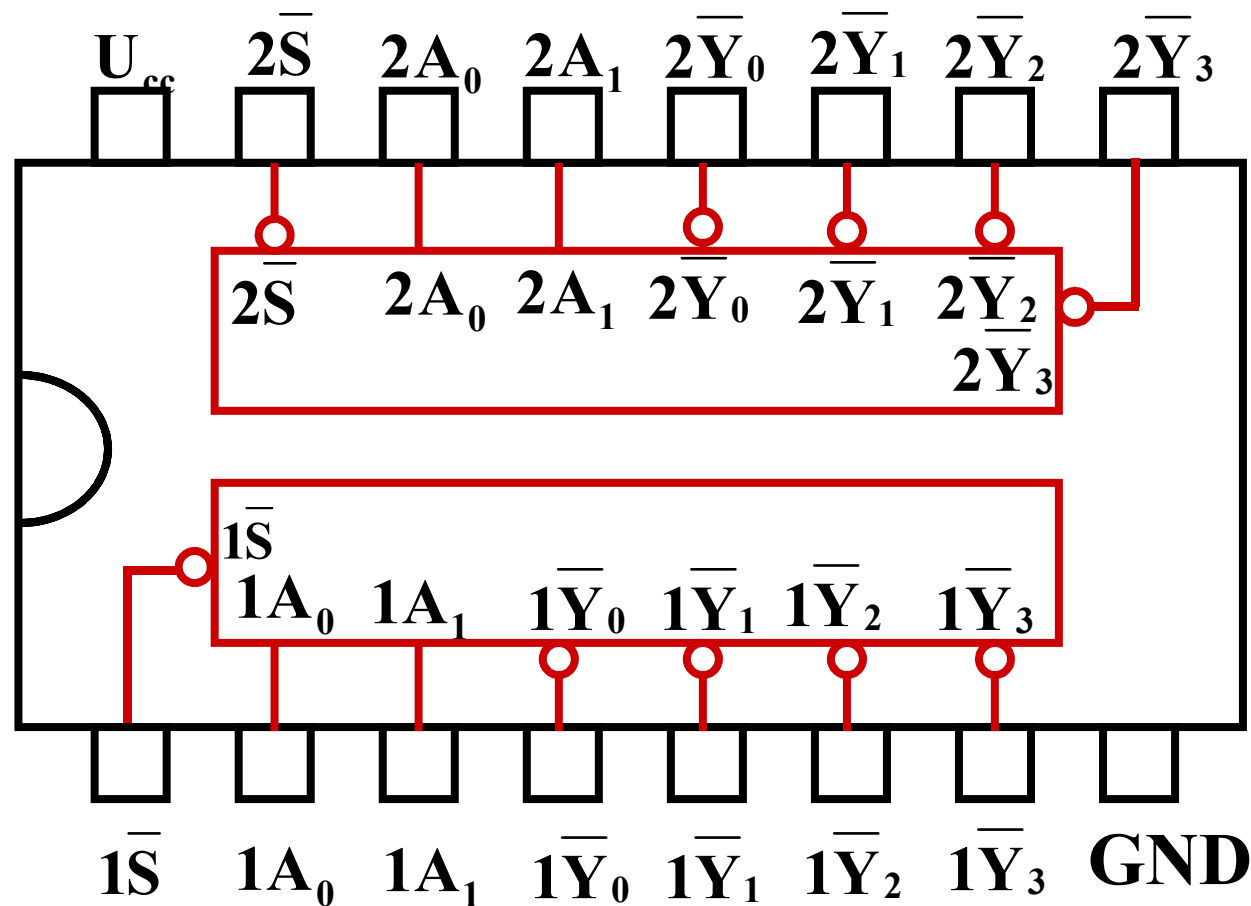


74LS139的功能表

| \bar{S} | A_1 | A_0 | \bar{Y}_0 | \bar{Y}_1 | \bar{Y}_2 | \bar{Y}_3 |
|-----------|-------|-------|-------------|-------------|-------------|-------------|
| 1 | X | X | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |

“ $\bar{}$ ”表示低电平有效。

74LS139管脚图



一片139种含两个2-4译码器

3位二进制译码器(3线-8线译码器)

| A_2 | A_1 | A_0 | Y_7 | Y_6 | Y_5 | Y_4 | Y_3 | Y_2 | Y_1 | Y_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

输入：3位二进制代码

输出：8个互斥的信号（高电平有效）

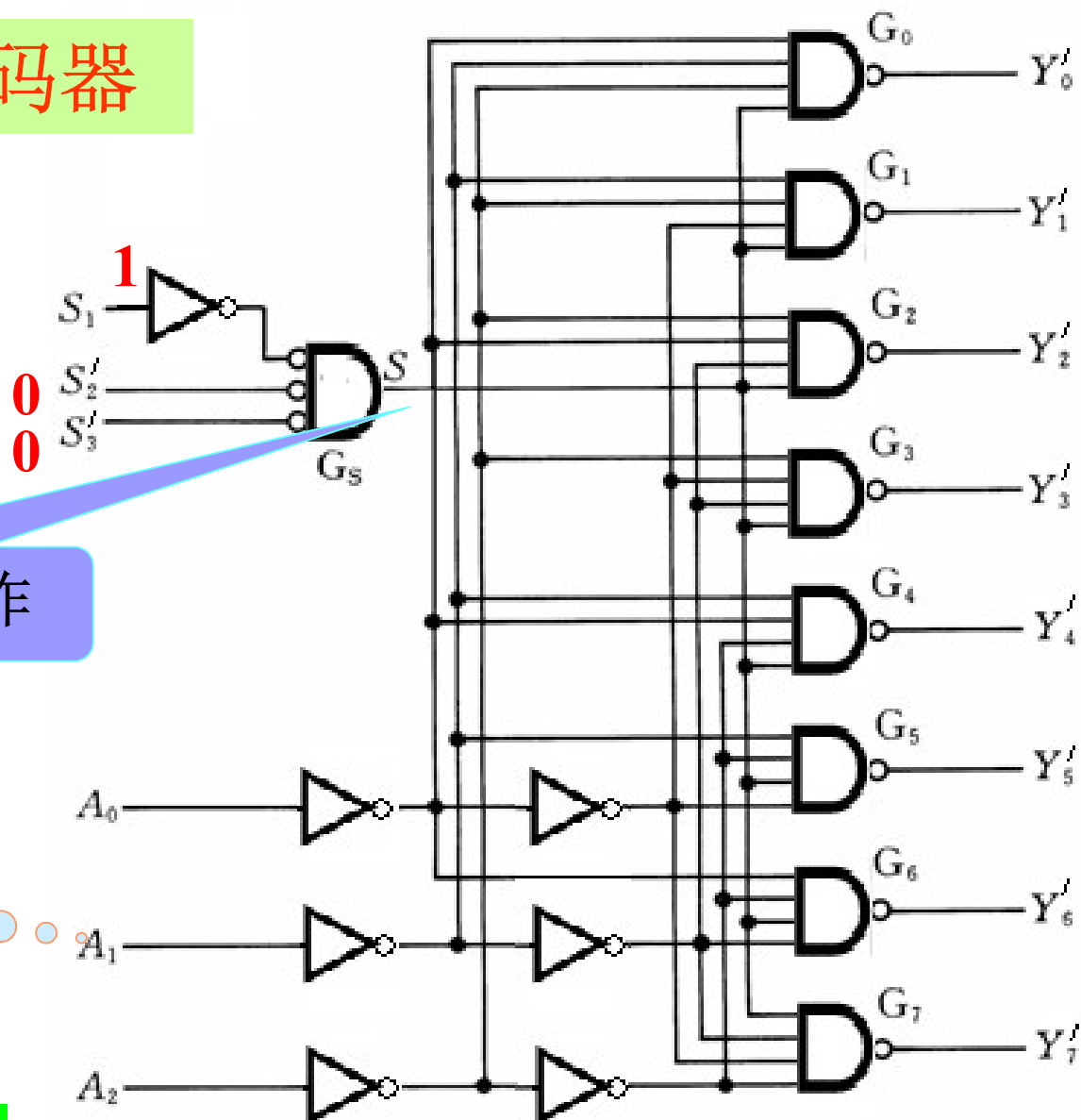
74HC138集成译码器

片选输入端
(使能端)

$S=1$, 译码器正常工作

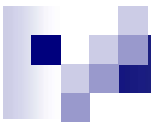
地址输入端

输出低电平有效



3线—8线译码器74HC138功能表

| 输 入 | | 输 出 | | | | | | | | | | |
|-------|---------------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| 使 能 | 选 择 | | | | | | | | | | | |
| S_1 | $S'_2 + S'_3$ | A_2 | A_1 | A_0 | Y'_0 | Y'_1 | Y'_2 | Y'_3 | Y'_4 | Y'_5 | Y'_6 | Y'_7 |
| 0 | × | × | × | × | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| × | 1 | × | × | × | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |



当 $S_1=1, S_2=0, S_3=0$ （即 $S=1$ ）时，可得输出

$$Y'_0 = (A'_2 A'_1 A'_0)' = m'_0$$

$$Y'_4 = (A_2 A'_1 A'_0)' = m'_4$$

$$Y'_1 = (A'_2 A'_1 A_0)' = m'_1$$

$$Y'_5 = (A_2 A'_1 A_0)' = m'_5$$

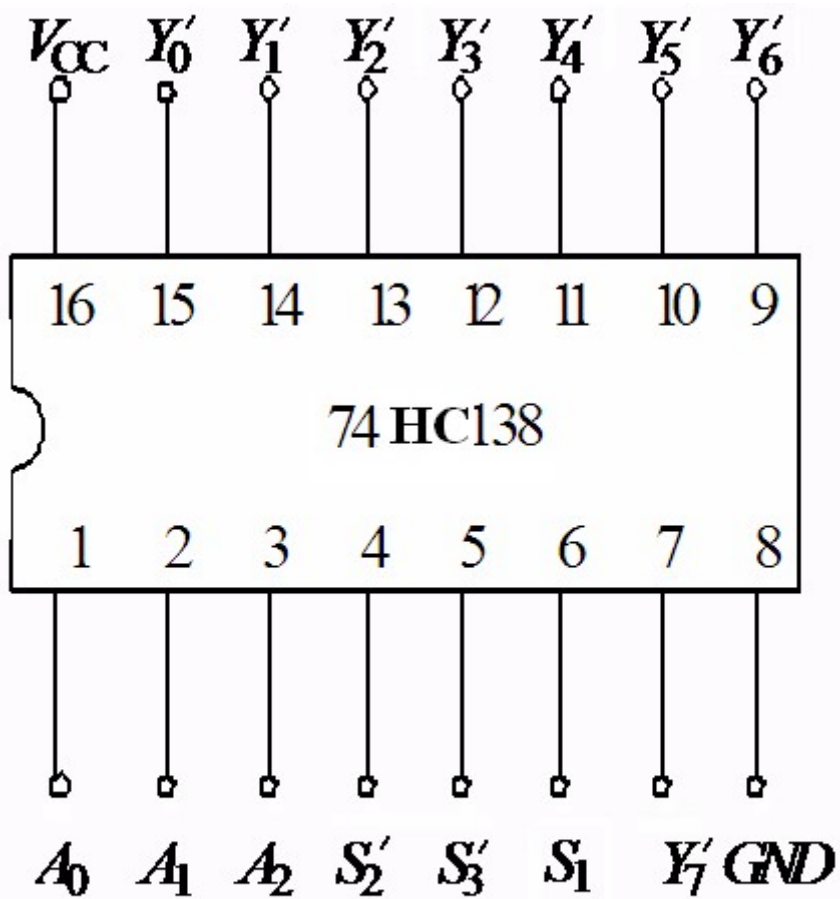
$$Y'_2 = (A'_2 A_1 A'_0)' = m'_2$$

$$Y'_6 = (A_2 A_1 A'_0)' = m'_6$$

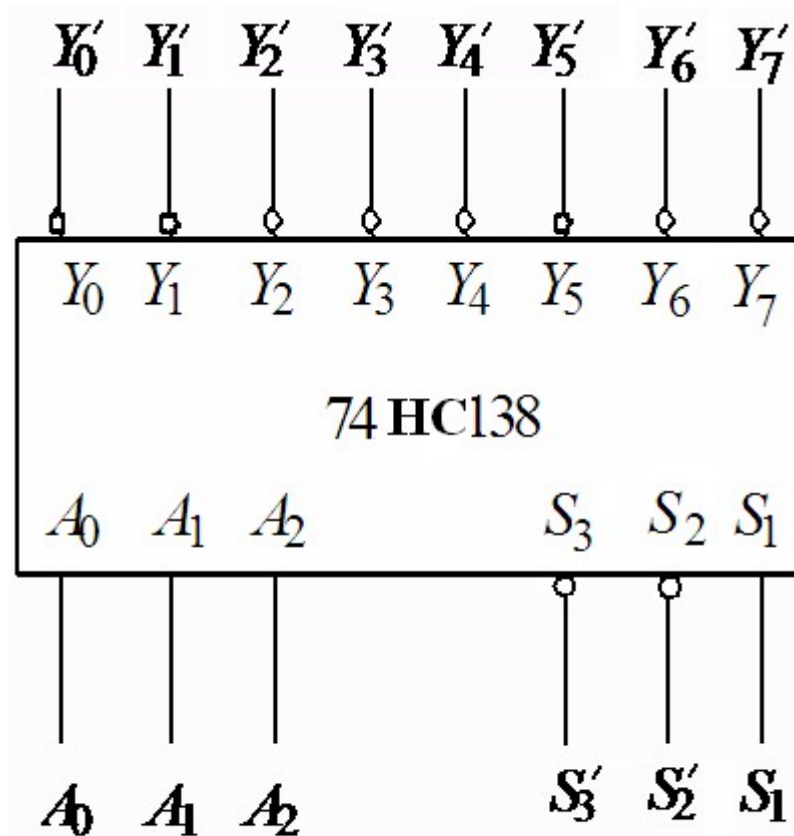
$$Y'_3 = (A'_2 A_1 A_0)' = m'_3$$

$$Y'_7 = (A_2 A_1 A_0)' = m'_7$$



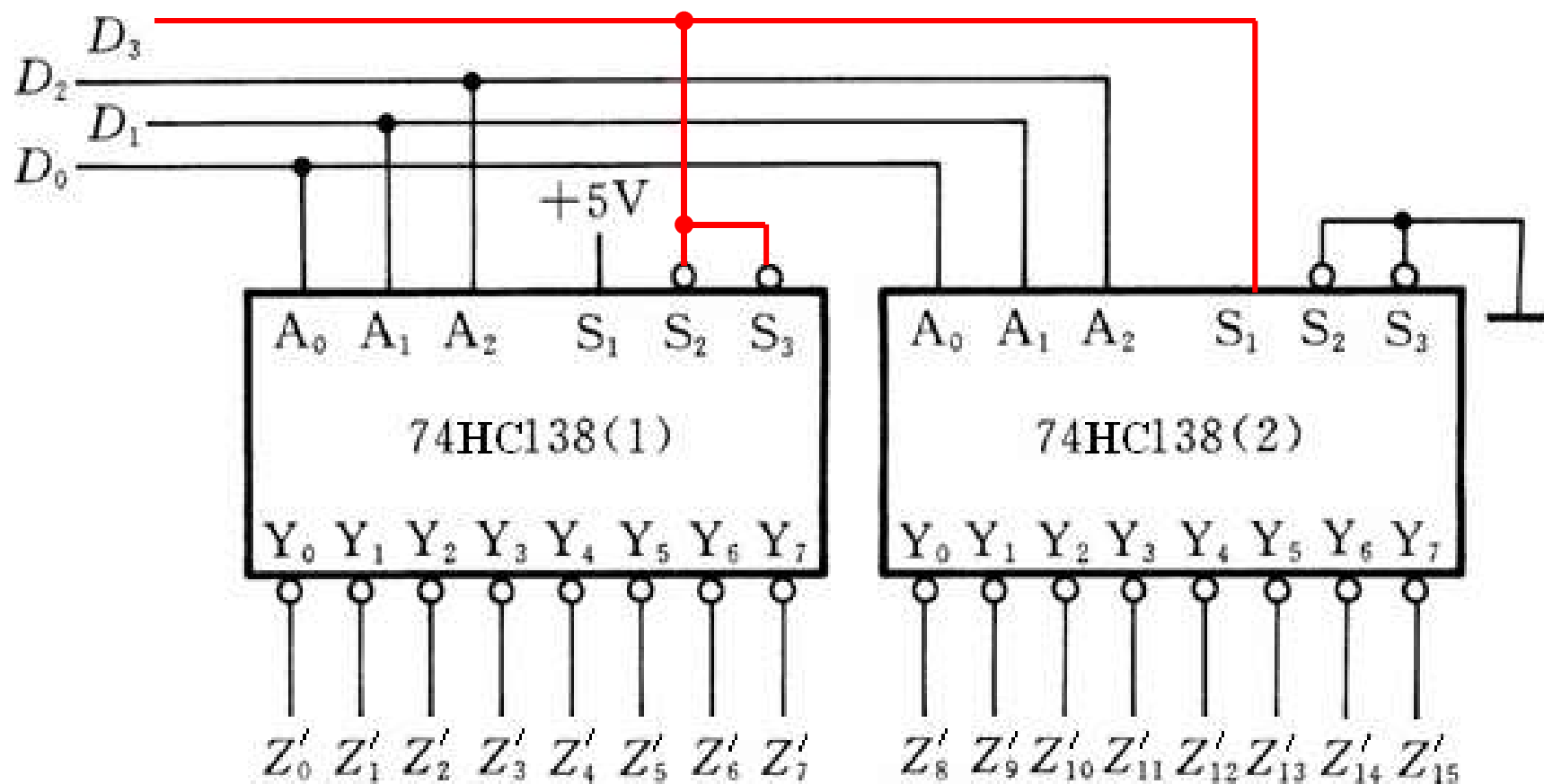


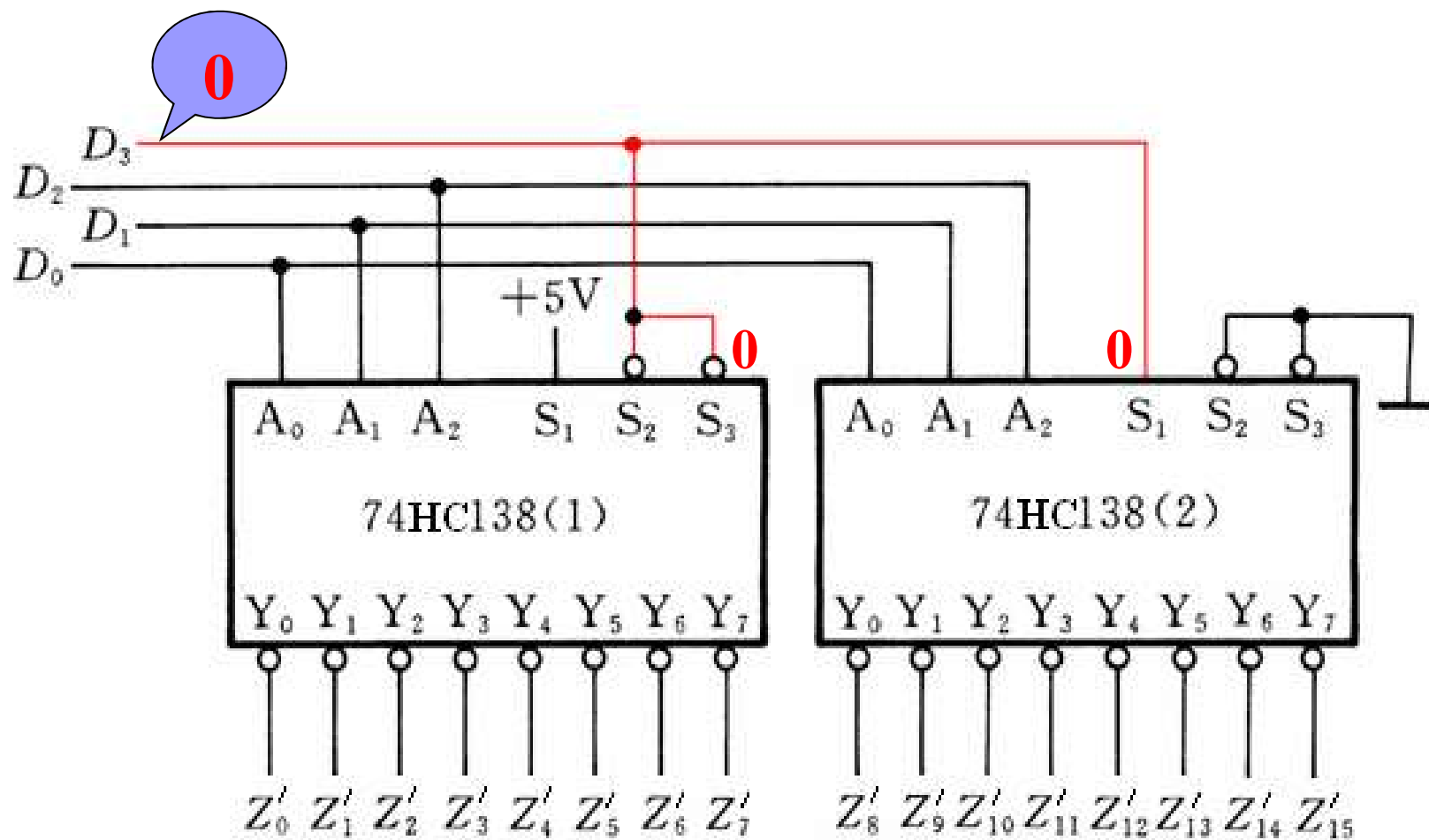
(a) 引脚排列图



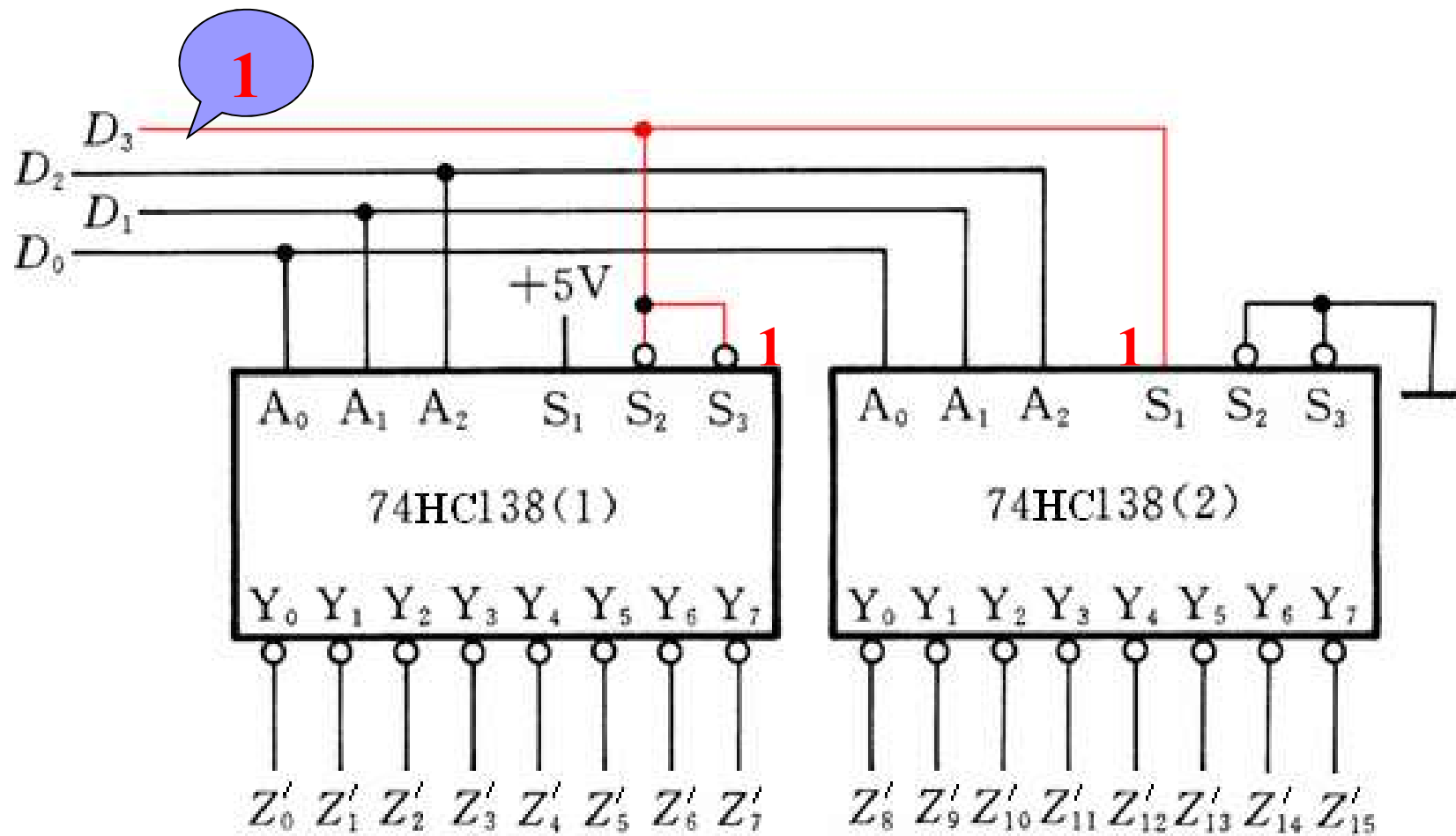
(b) 逻辑功能示意图

例：试用两片3线—8线译码器74HC138组成4线—16线译码器。



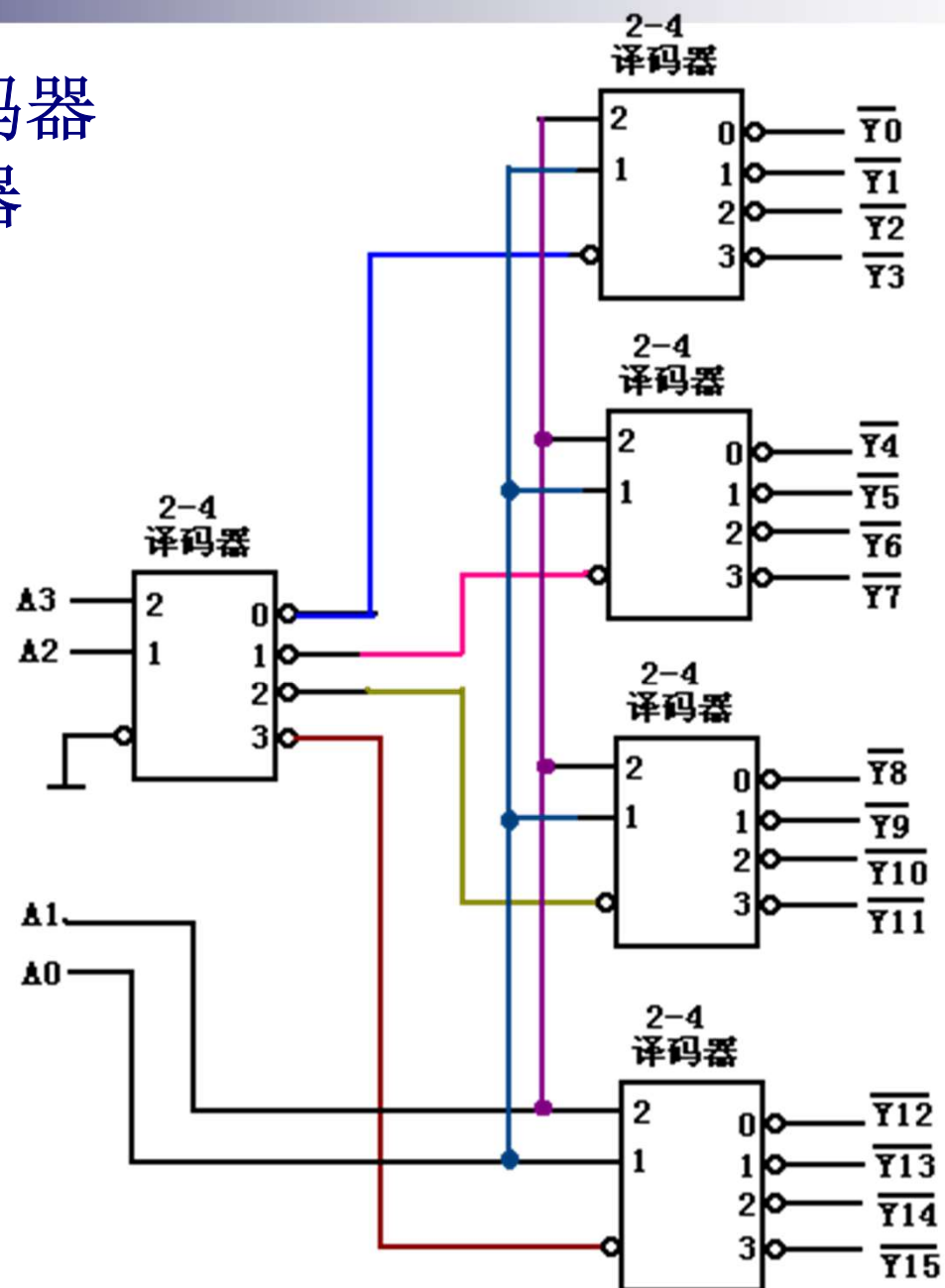


(1)片工作，(2)片禁止。若输入 $D_3D_2D_1D_0=0100$ 时，译码器(1)输出11110111。

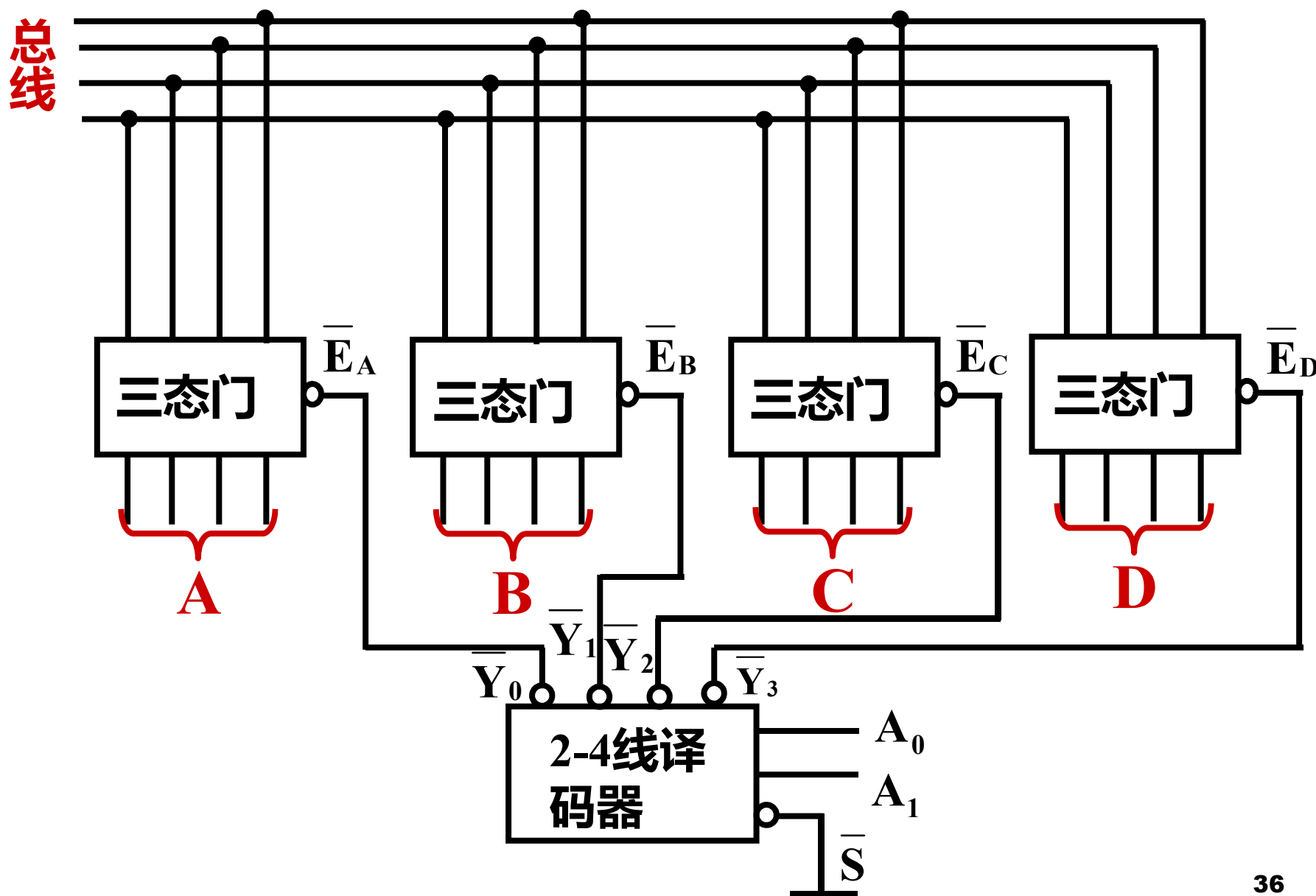


(2)片工作，(1)片禁止。若输入 $D_3D_2D_1D_0=1101$ 时，译码器(2)输出1111011。

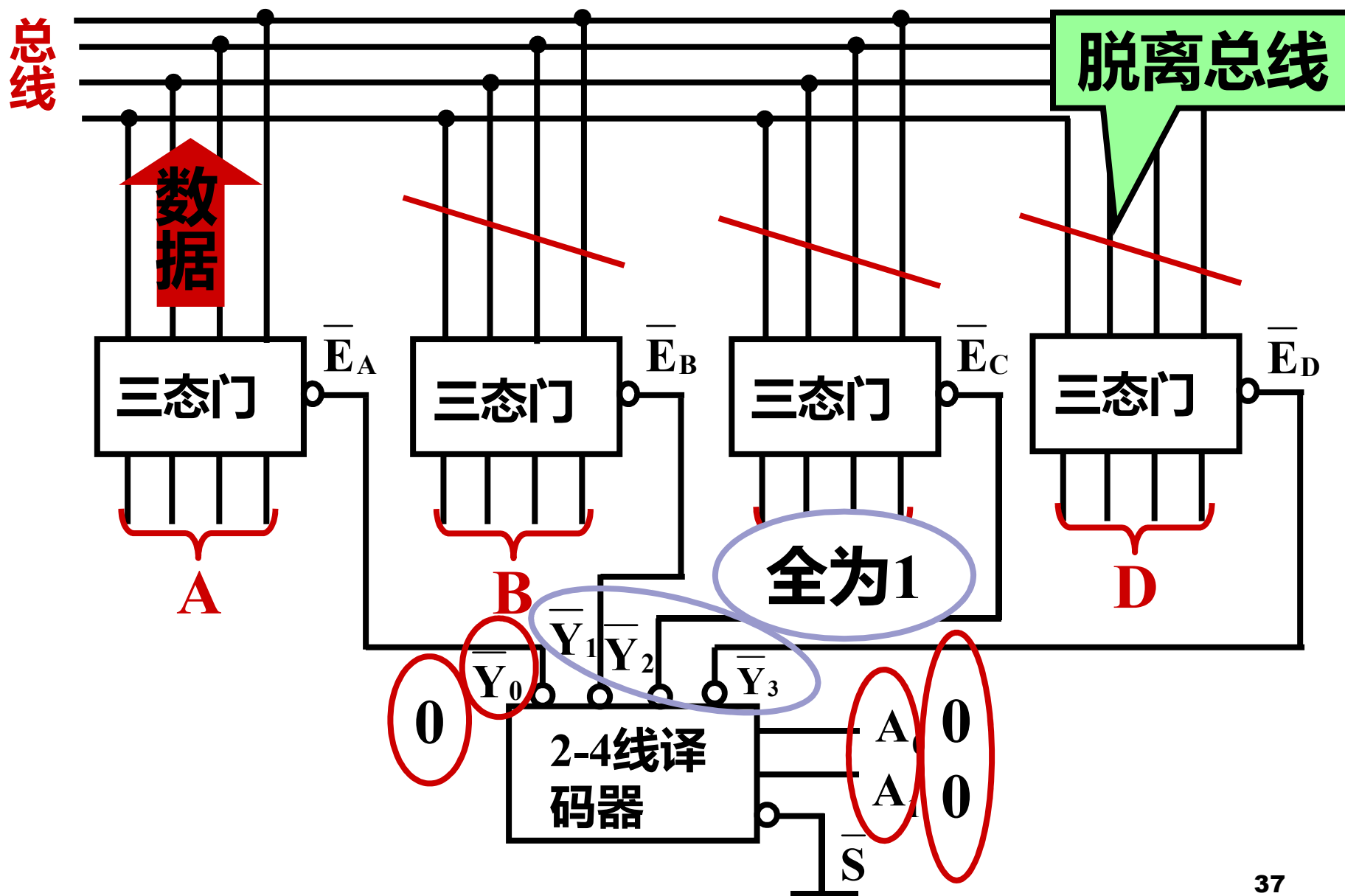
例：5片2-4线译码器
构成4-16线译码器



应用：利用线译码器分时将采样数据送入计算机。



工作原理：（以 $A_0A_1=00$ 为例）



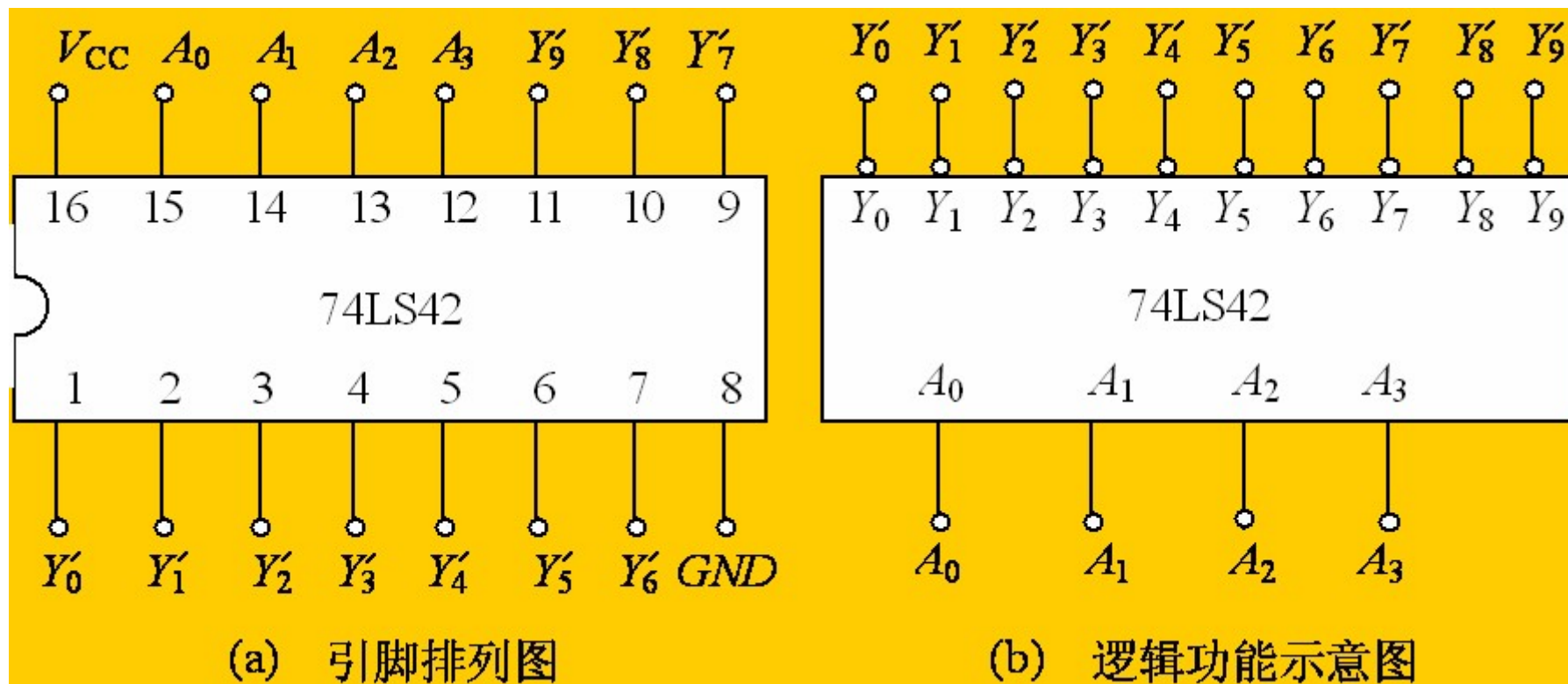
二、BCD—十进制译码器(BCD-to-Decimal Decoder)

输入端：4 输出端：10

BCD—十(二—十)进制译码器的输入是十进制数的4位二进制编码 (BCD码)，分别用 A_3 、 A_2 、 A_1 、 A_0 表示；输出的是与10个十进制数字相对应的10个信号，用 $Y_9 \sim Y_0$ 表示。

由于二—十进制译码器有4根输入线，10根输出线，所以又称为4线—10线译码器。

集成8421 BCD码译码器74LS42



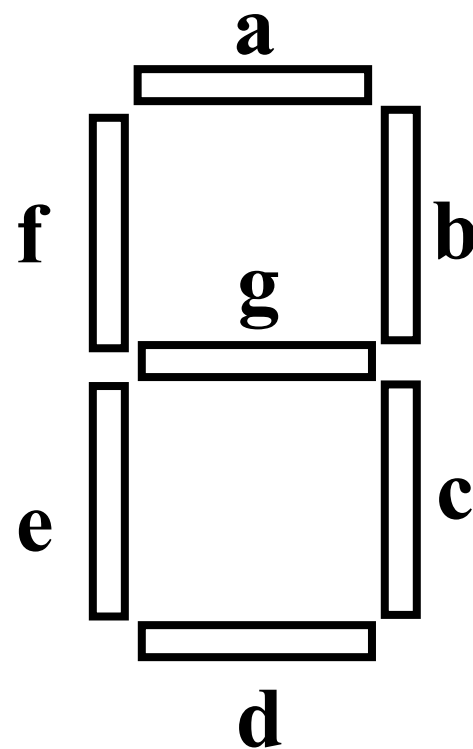
三、显示译码器

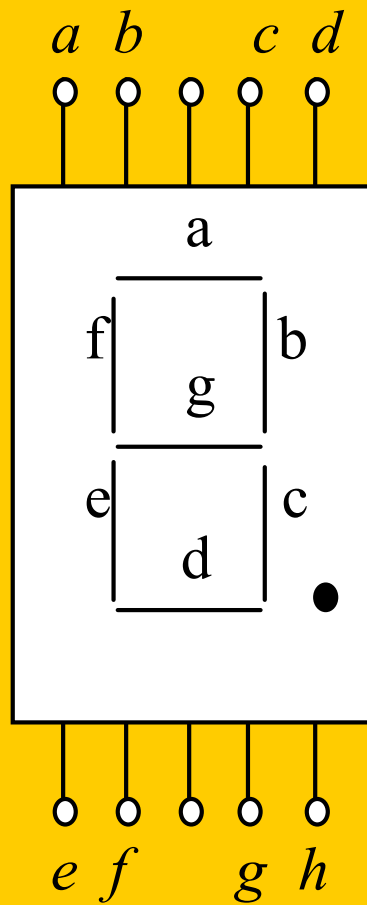
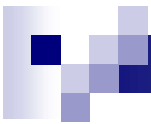
用来驱动各种显示器件，从而将用二进制代码表示的数字、文字、符号翻译成人们习惯的形式直观地显示出来的电路，称为**显示译码器**。



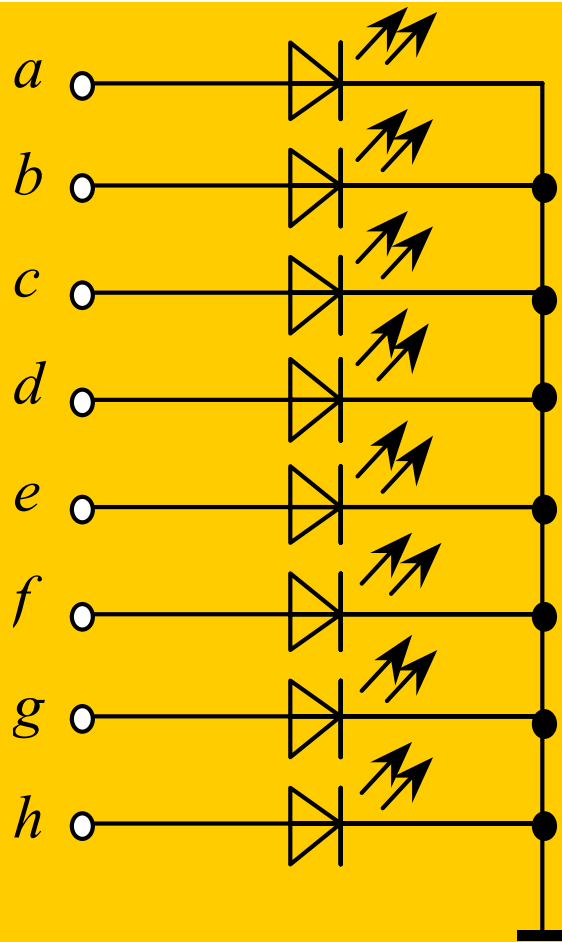
半导体数码管

显示器件： 常用的是**七段显示器件**

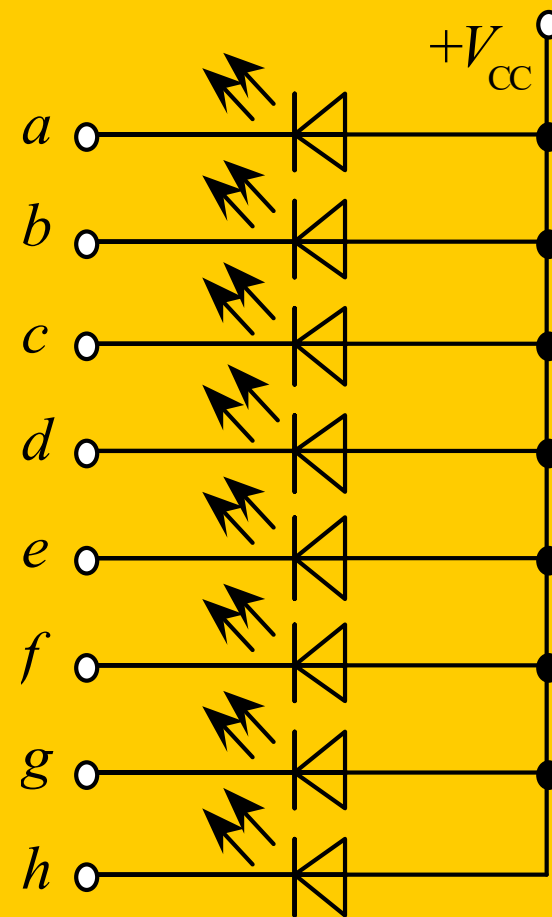




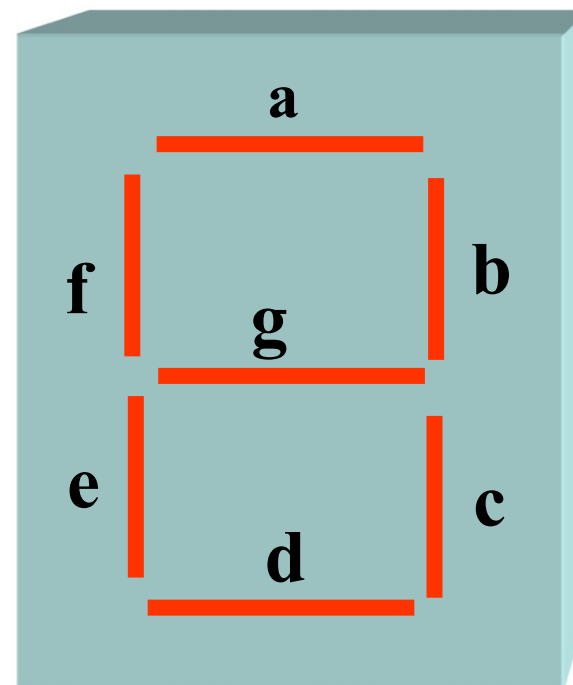
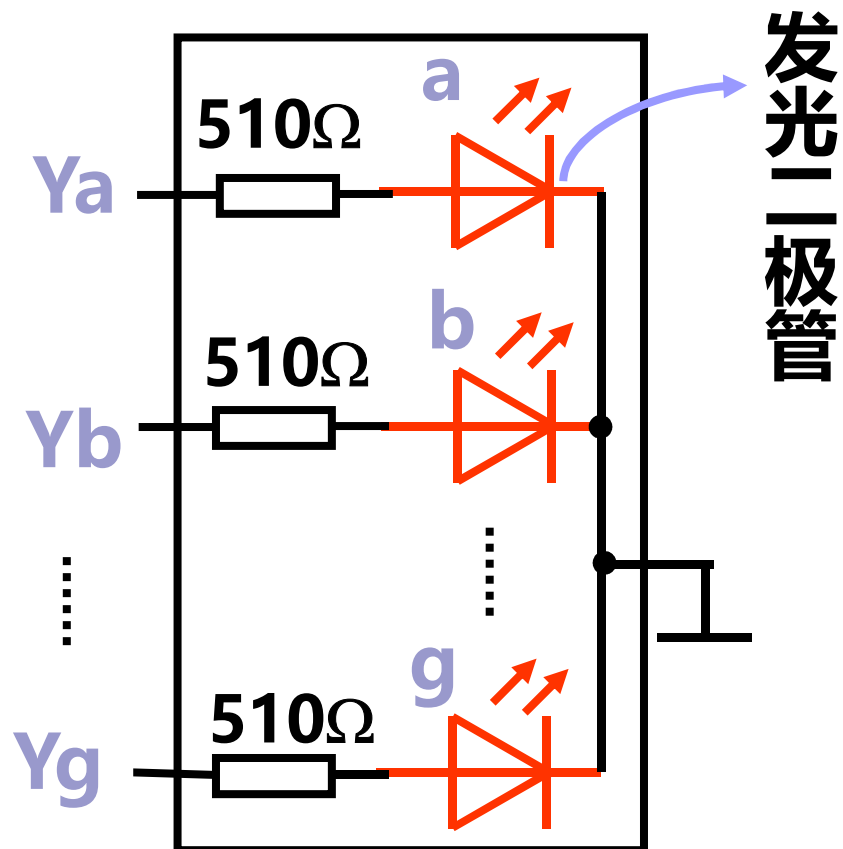
(a) 外形图



(b) 共阴极



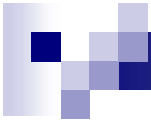
(c) 共阳极



Ya-Yg: 控制信号

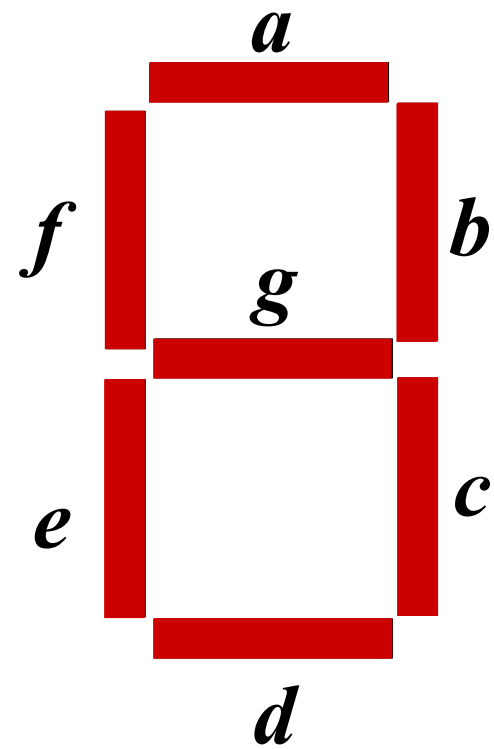
高电平时,对应的LED亮

低电平时,对应的LED灭



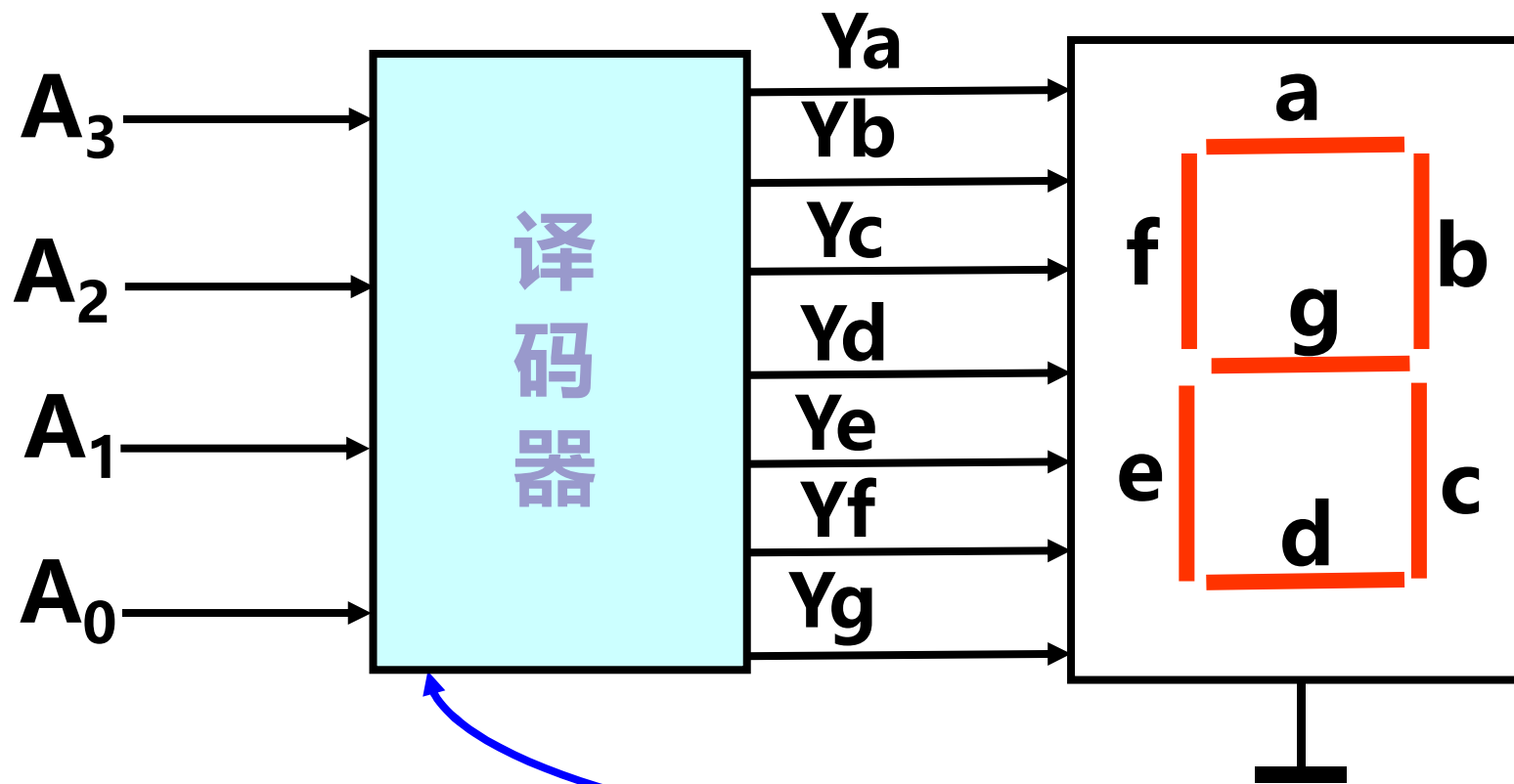
| <i>a</i> | <i>b</i> | <i>c</i> | <i>d</i> | <i>e</i> | <i>f</i> | <i>g</i> |
|----------|----------|----------|----------|----------|----------|----------|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |

⋮



BCD—七段显示译码器 (BCD-to-7-segment decoder)

A_3-A_0 : 输入数据



要设计的七段显示译码器

| 十进制数 | A ₃ A ₂ A ₁ A ₀ | Y _a Y _b Y _c Y _d Y _e Y _f Y _g | 显示字形 |
|------|---|--|------|
| 0 | 0 0 0 0 | 1 1 1 1 1 1 0 | 0 |
| 1 | 0 0 0 1 | 0 1 1 0 0 0 0 | 1 |
| 2 | 0 0 1 0 | 1 1 0 1 1 0 1 | 2 |
| 3 | 0 0 1 1 | 1 1 1 1 0 0 1 | 3 |
| 4 | 0 1 0 0 | 0 1 1 0 0 1 1 | 4 |
| 5 | 0 1 0 1 | 1 0 1 1 0 1 1 | 5 |
| 6 | 0 1 1 0 | 0 0 1 1 1 1 1 | 6 |
| 7 | 0 1 1 1 | 1 1 1 0 0 0 0 | 7 |
| 8 | 1 0 0 0 | 1 1 1 1 1 1 1 | 8 |
| 9 | 1 0 0 1 | 1 1 1 0 0 1 1 | 9 |

先设计输出Ya的逻辑表示式及电路图

Ya

A_1A_0

A_3A_2

| | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 | 1 | 0 | 1 | 1 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 1 | 0 | 0 |
| 10 | 1 | 1 | 0 | 0 |

$$Y_a = (A'_3A'_2A'_1A_0 + A_3A_1 + A_2A'_0)'$$

七段显示译码器7448引脚排列图

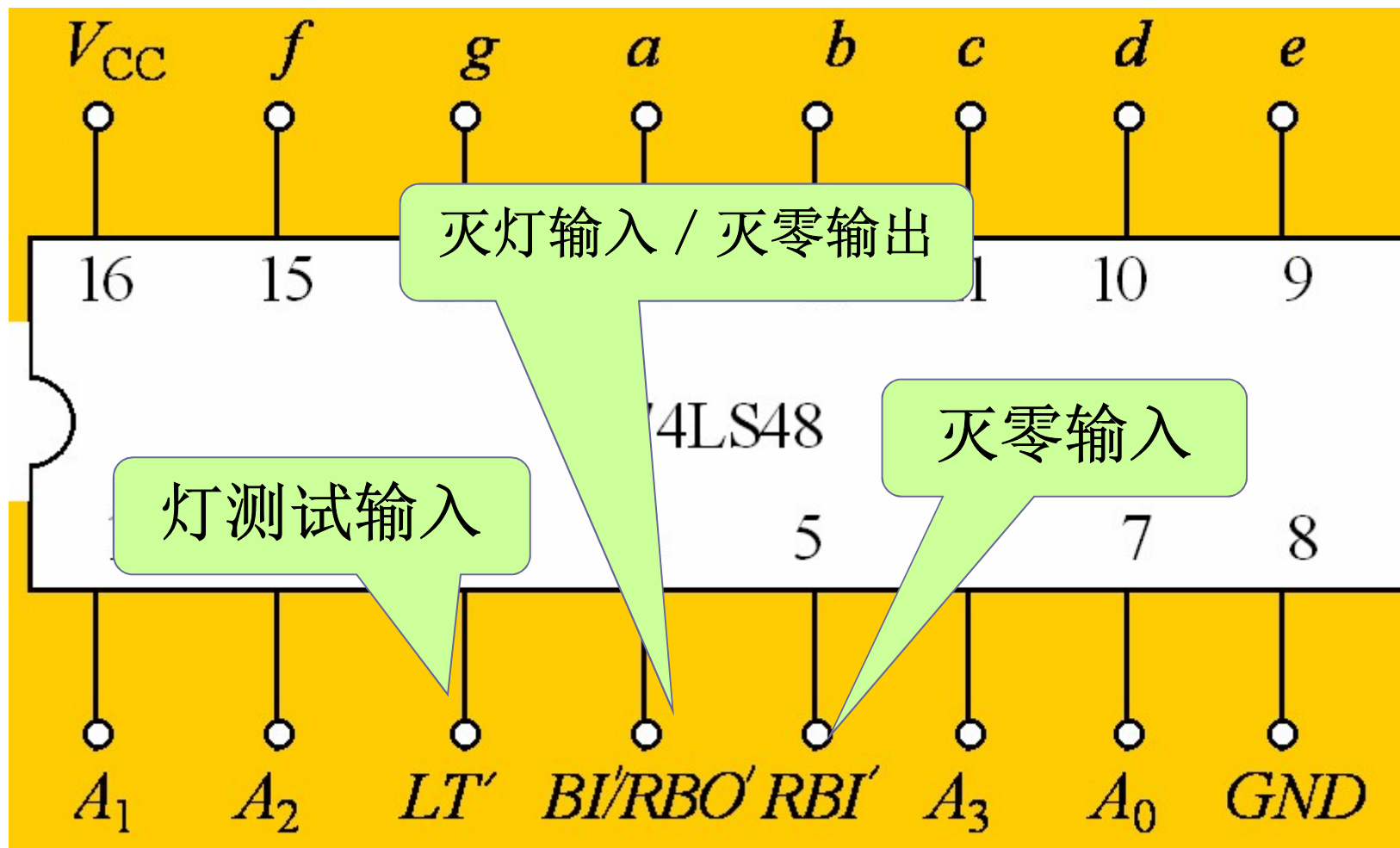
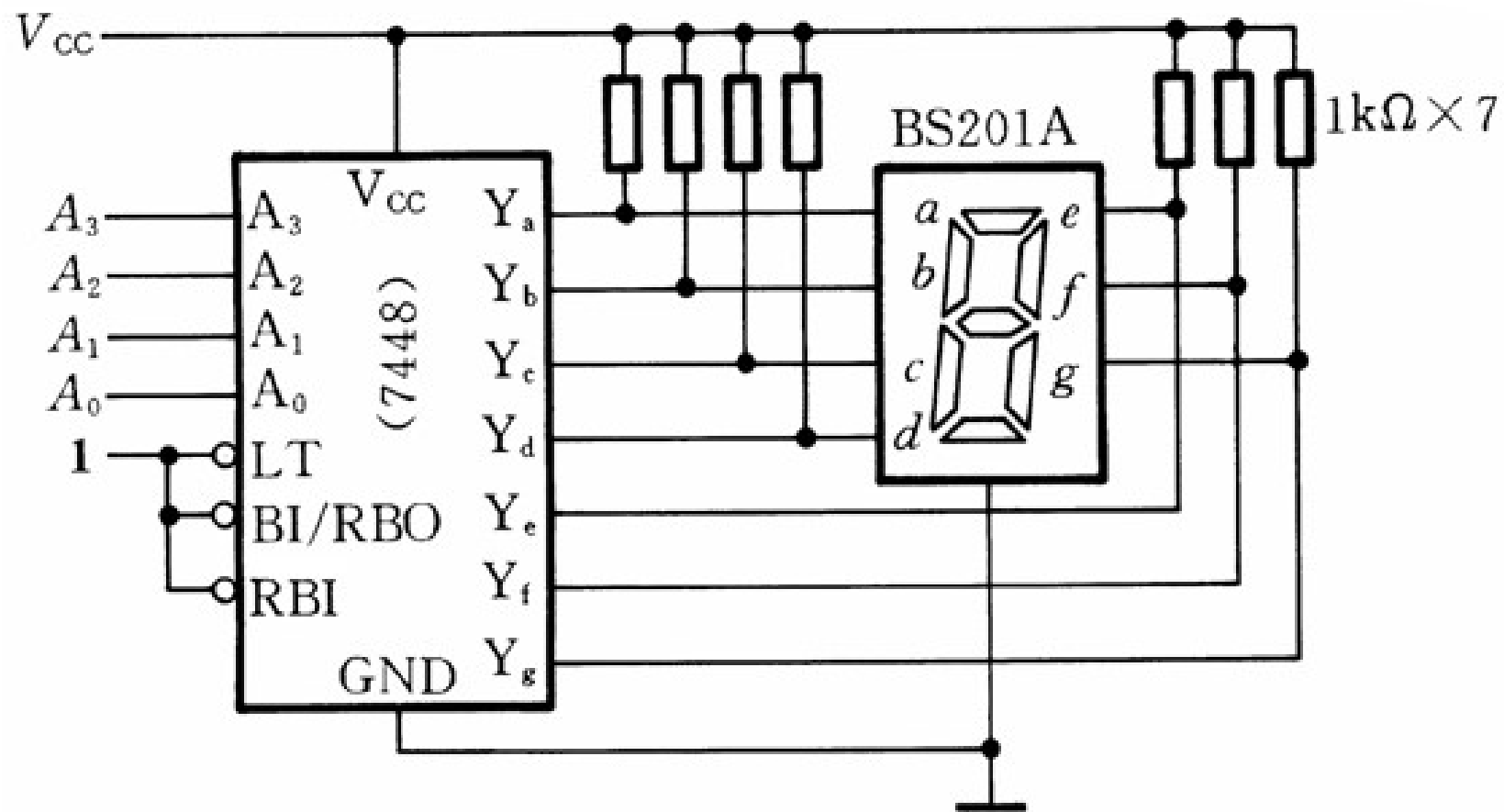
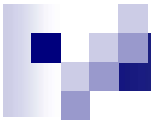


图4.3.18 用7448驱动BS201的连接方法





RBI和RBO配合使用，可使多位数字显示时的最高位及小数点后最低位的0不显示

| | | | | | | | | |
|--|--|--|---|---|---|---|--|--|
| | | | 6 | 7 | . | 9 | | |
|--|--|--|---|---|---|---|--|--|

§ 4.3.3 数据分配器与数据选择器

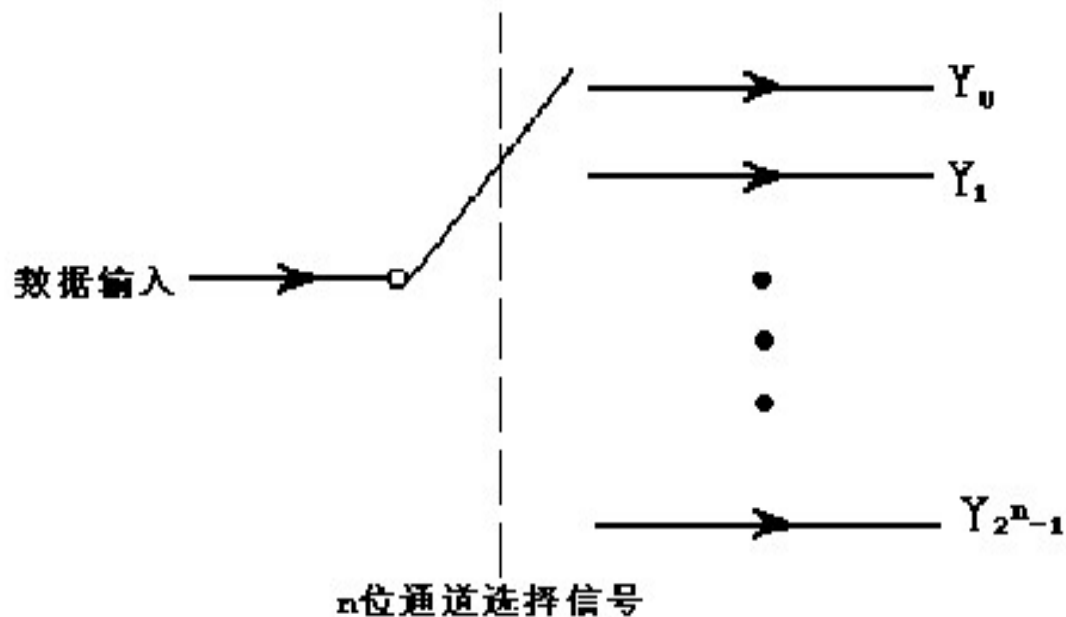
一、数据分配器

定义： 将公共数据线上的信号根据需要送到多个不同通道上去的逻辑电路。

输入端:1个

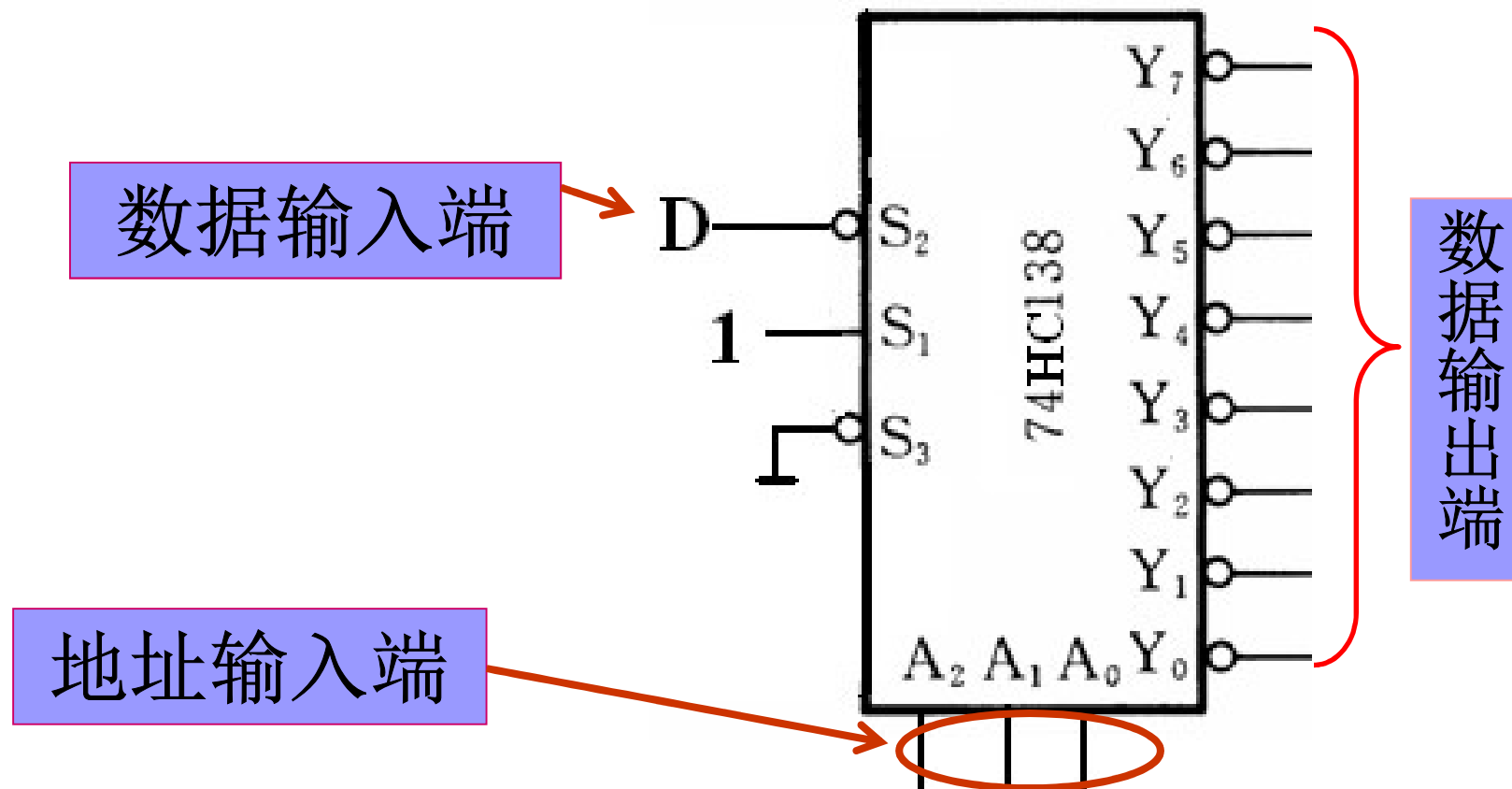
输出端: 2^n 个

框图:



数据分配器示意图

由74HC138构成的1路-8路数据分配器

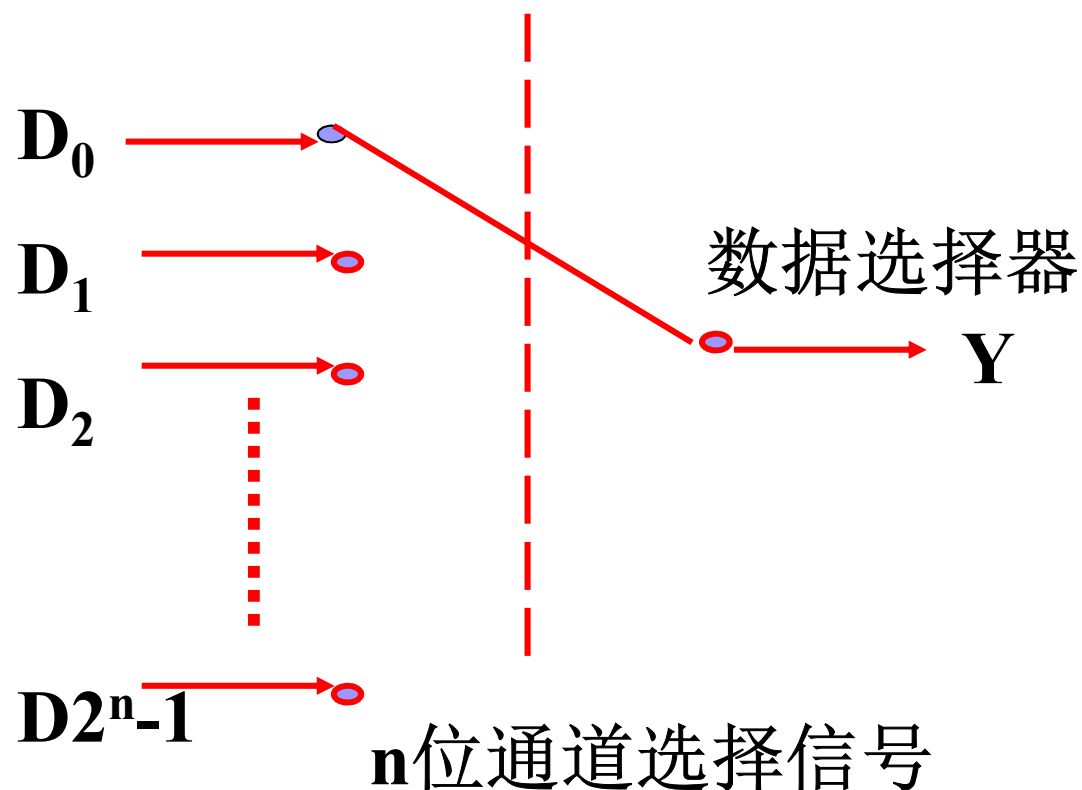


二、数据选择器

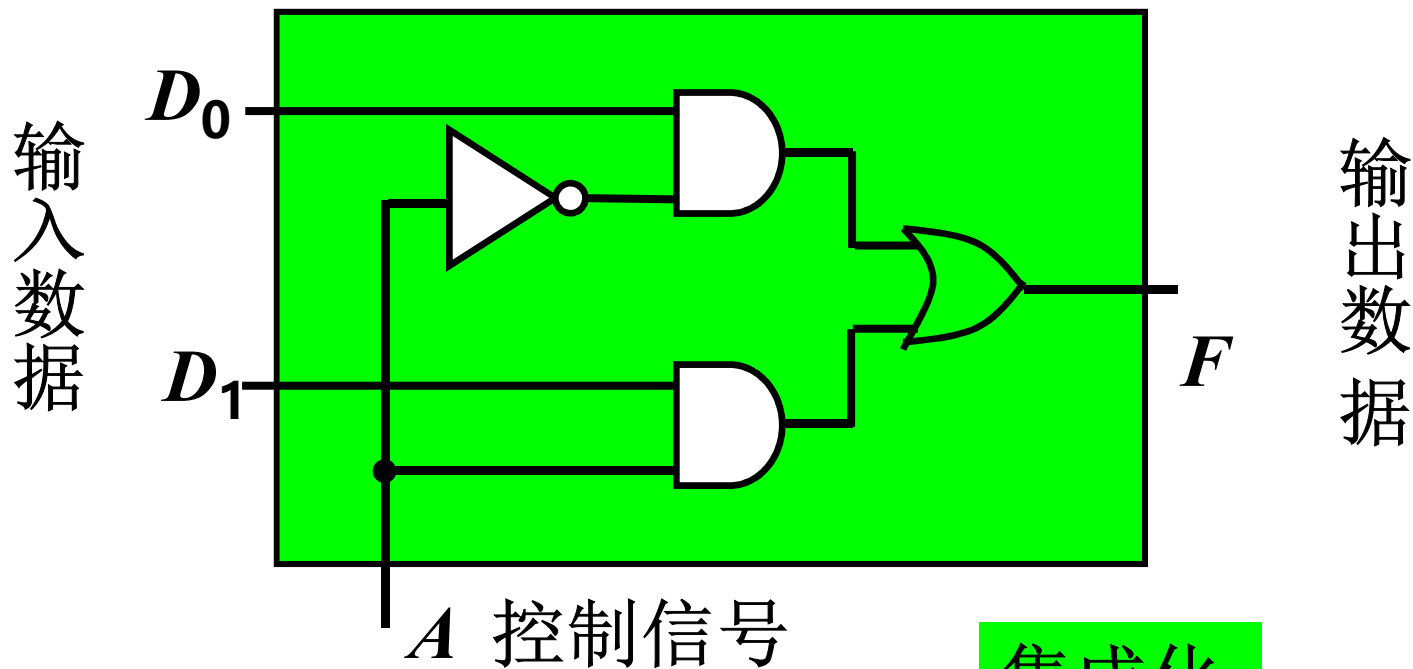
定义： 根据需要将多路信号中选择一路送到公共数据线上的逻辑电路(又称多路开关).

输入端： 2^n 个

输出端： 1 个



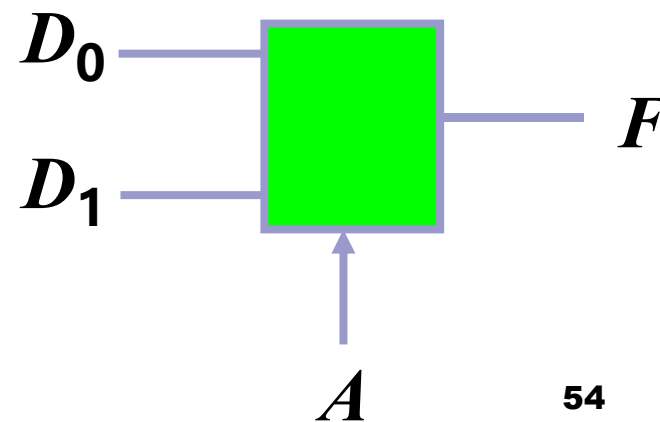
1、2选1数据选择器



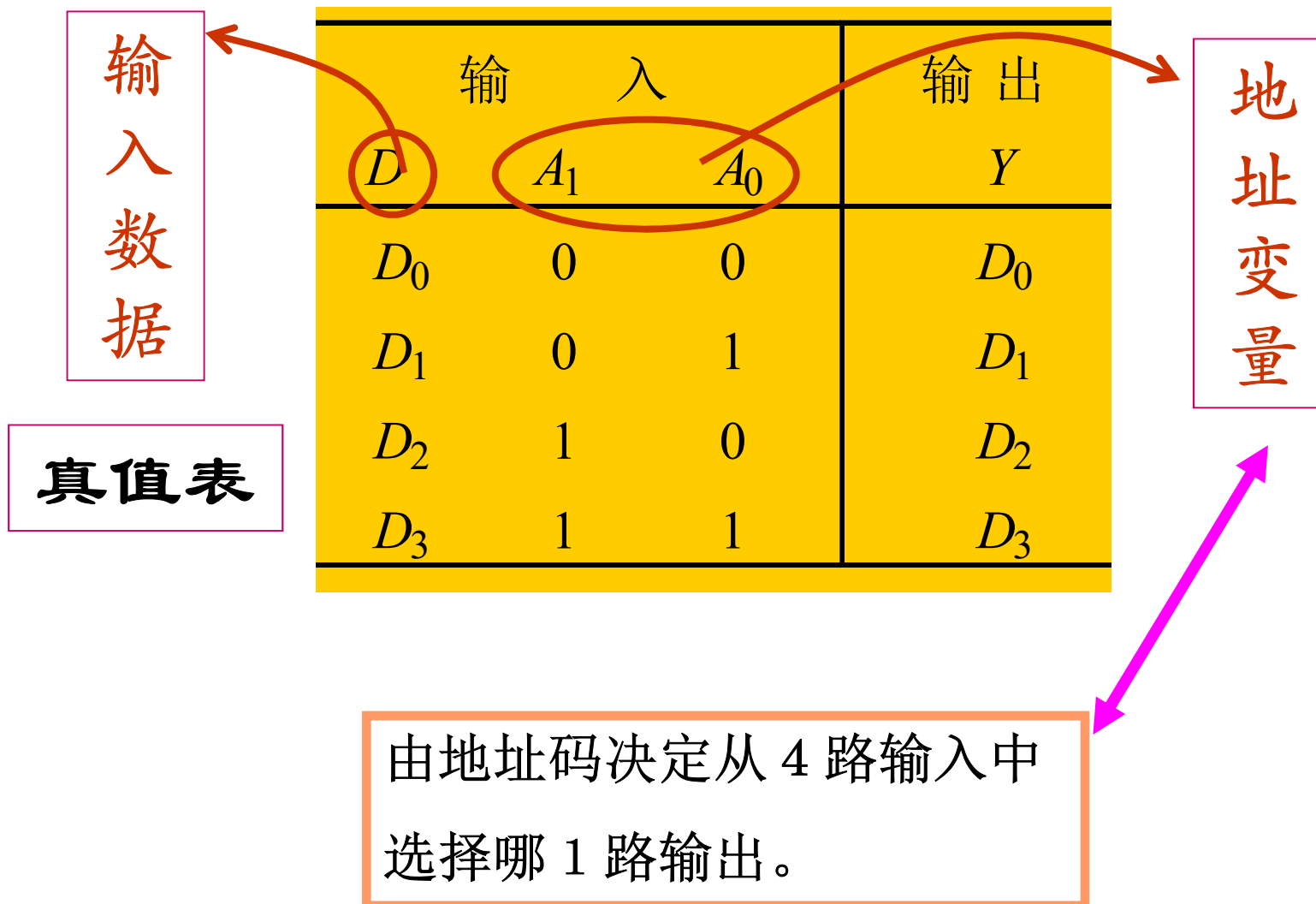
集成化

$$F = A'D_0 + AD_1$$

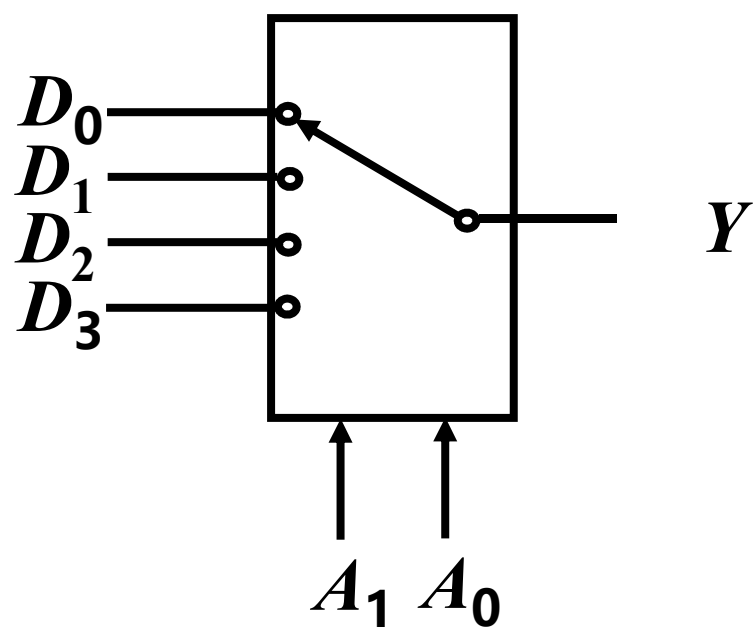
| A | F |
|-----|-------|
| 0 | D_0 |
| 1 | D_1 |



2、4选1数据选择器



即：



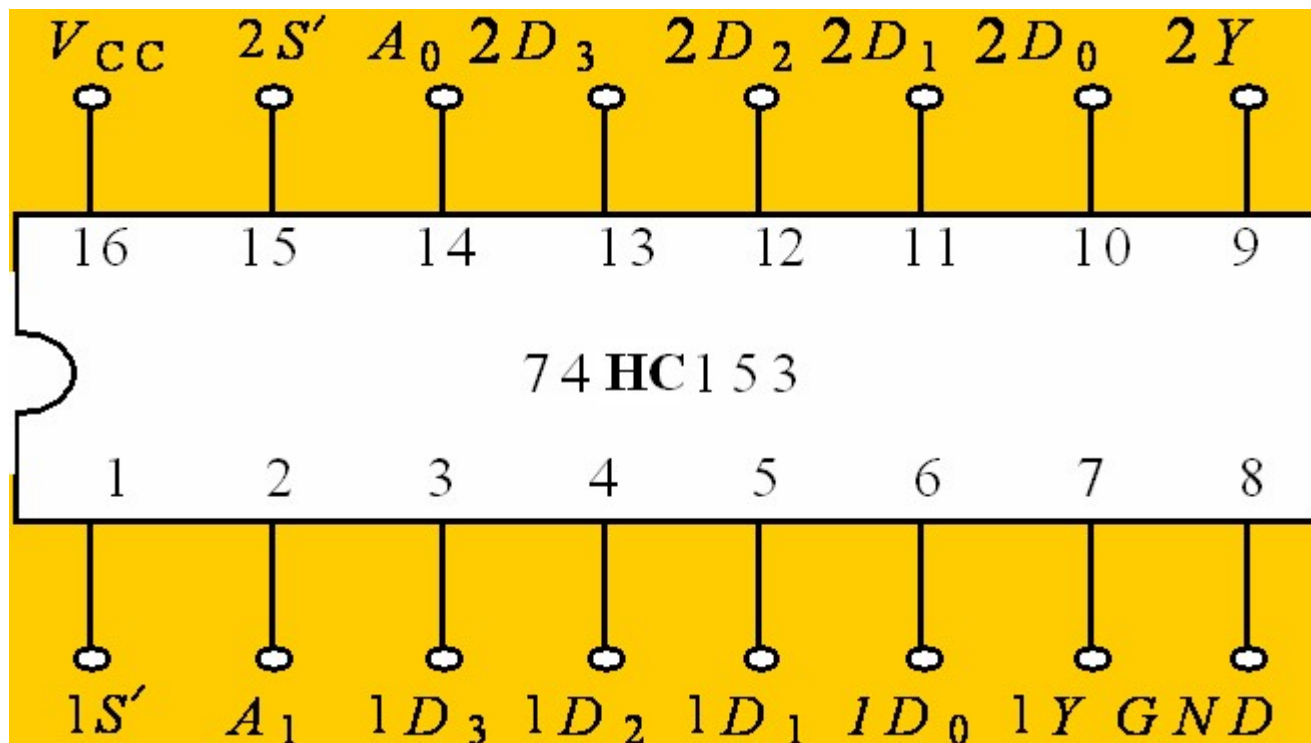
| A_1 | A_0 | Y |
|-------|-------|-------|
| 0 | 0 | D_0 |
| 0 | 1 | D_1 |
| 1 | 0 | D_2 |
| 1 | 1 | D_3 |

$$Y = A_1' A_0' D_0 + A_1' A_0 D_1 + A_1 A_0' D_2 + A_1 A_0 D_3$$

集成电路数据选择器

型号:74HC153

双4选1数据选择器



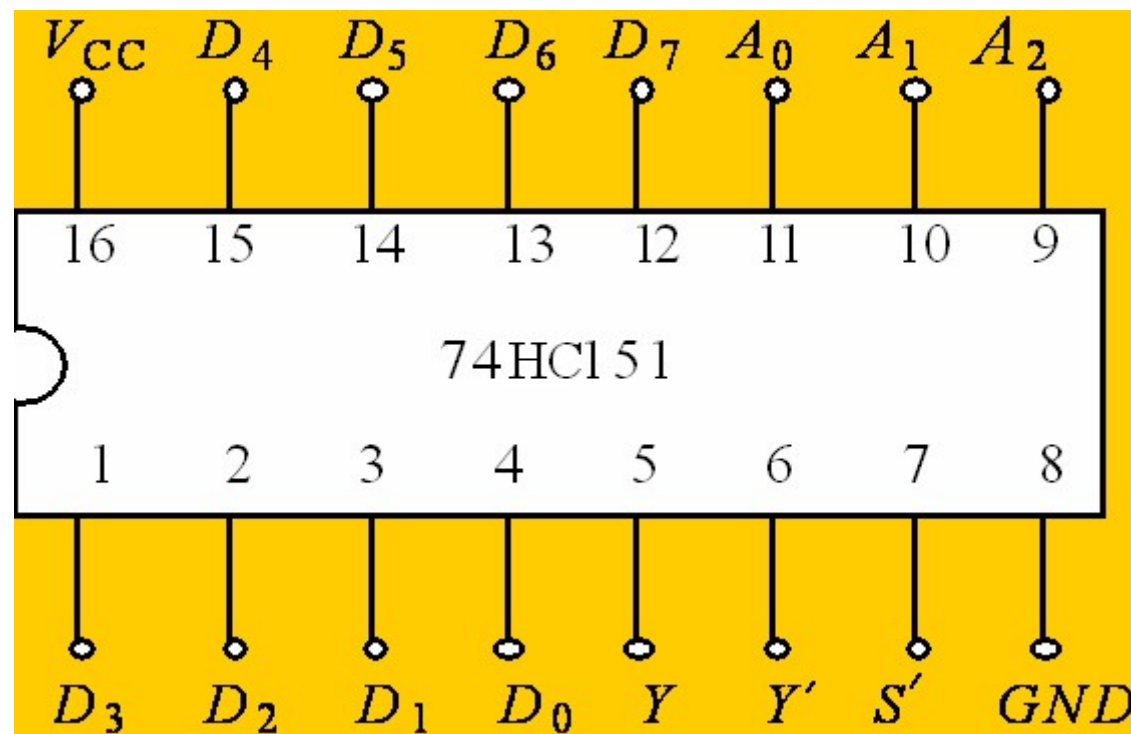
| 输入 | | | 输出 |
|--------|--------|--------|----------|
| A_1 | A_0 | S'_1 | Y_1 |
| ϕ | ϕ | 1 | 0 |
| 0 | 0 | 0 | D_{10} |
| 0 | 1 | 0 | D_{11} |
| 1 | 0 | 0 | D_{12} |
| 1 | 1 | 0 | D_{13} |

$$Y_1 = [(A'_1 A'_0) D_{10} + (A'_1 A_0) D_{11} + (A_1 A'_0) D_{12} + (A_1 A_0) D_{13}] \cdot S_1$$

$$Y_2 = [(A'_1 A'_0) D_{20} + (A'_1 A_0) D_{21} + (A_1 A'_0) D_{22} + (A_1 A_0) D_{23}] \cdot S_2$$



**集成8选
1数据选
择器
74HC151**



$S' = 1$ 时，选择器被禁止，无论地址码是什么， Y 总是等于 0

$S' = 0$

$$Y = (A_2' A_1' A_0') D_0 + (A_2' A_1' A_0) D_1 + (A_2' A_1 A_0') D_2 + (A_2' A_1 A_0) D_3 + (A_2 A_1' A_0') D_4 + (A_2 A_1' A_0) D_5 + (A_2 A_1 A_0') D_6 + (A_2 A_1 A_0) D_7$$

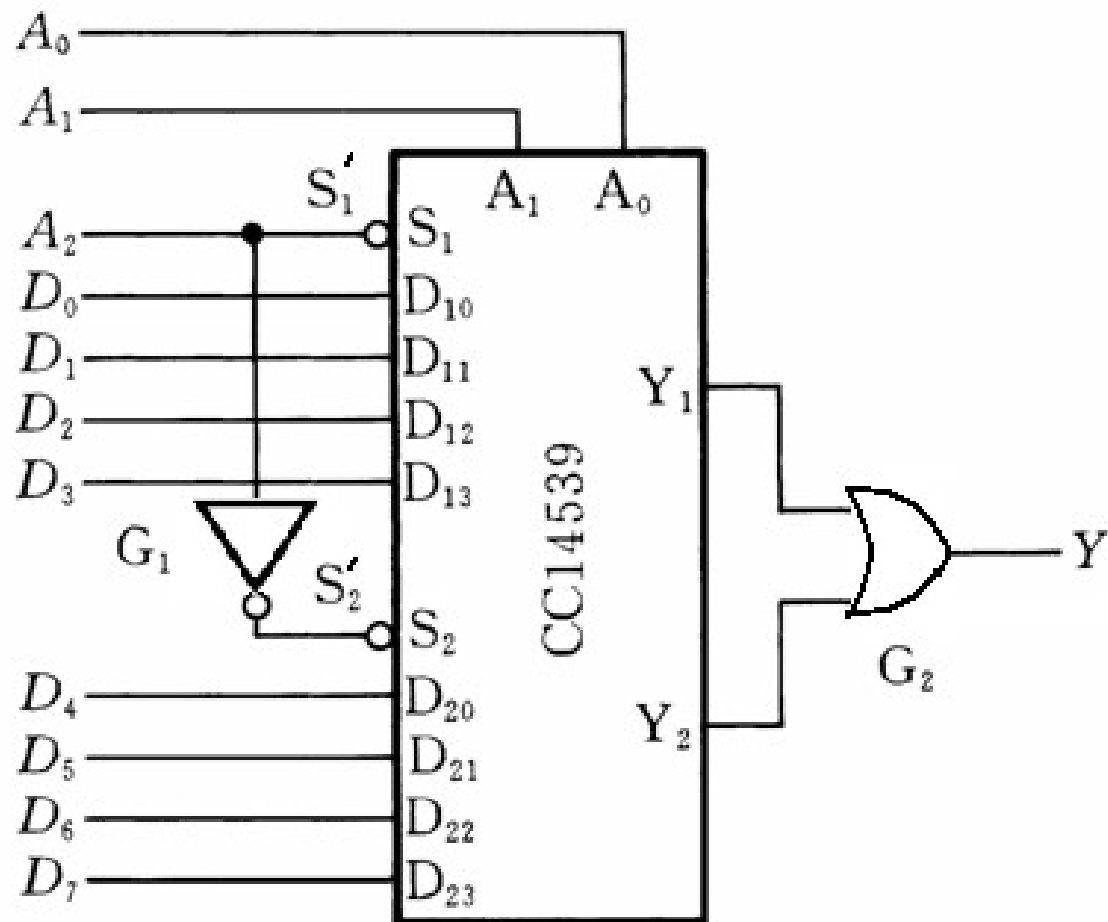
74HC151的真值表

| 输 入 | | | | | 输 出 | |
|----------|----------|----------|----------|------|-------|--------|
| D | A_2 | A_1 | A_0 | S' | Y | Y' |
| \times | \times | \times | \times | 1 | 0 | 1 |
| D_0 | 0 | 0 | 0 | 0 | D_0 | D'_0 |
| D_1 | 0 | 0 | 1 | 0 | D_1 | D'_1 |
| D_2 | 0 | 1 | 0 | 0 | D_2 | D'_2 |
| D_3 | 0 | 1 | 1 | 0 | D_3 | D'_3 |
| D_4 | 1 | 0 | 0 | 0 | D_4 | D'_4 |
| D_5 | 1 | 0 | 1 | 0 | D_5 | D'_5 |
| D_6 | 1 | 1 | 0 | 0 | D_6 | D'_6 |
| D_7 | 1 | 1 | 1 | 0 | D_7 | D'_7 |

扩展：

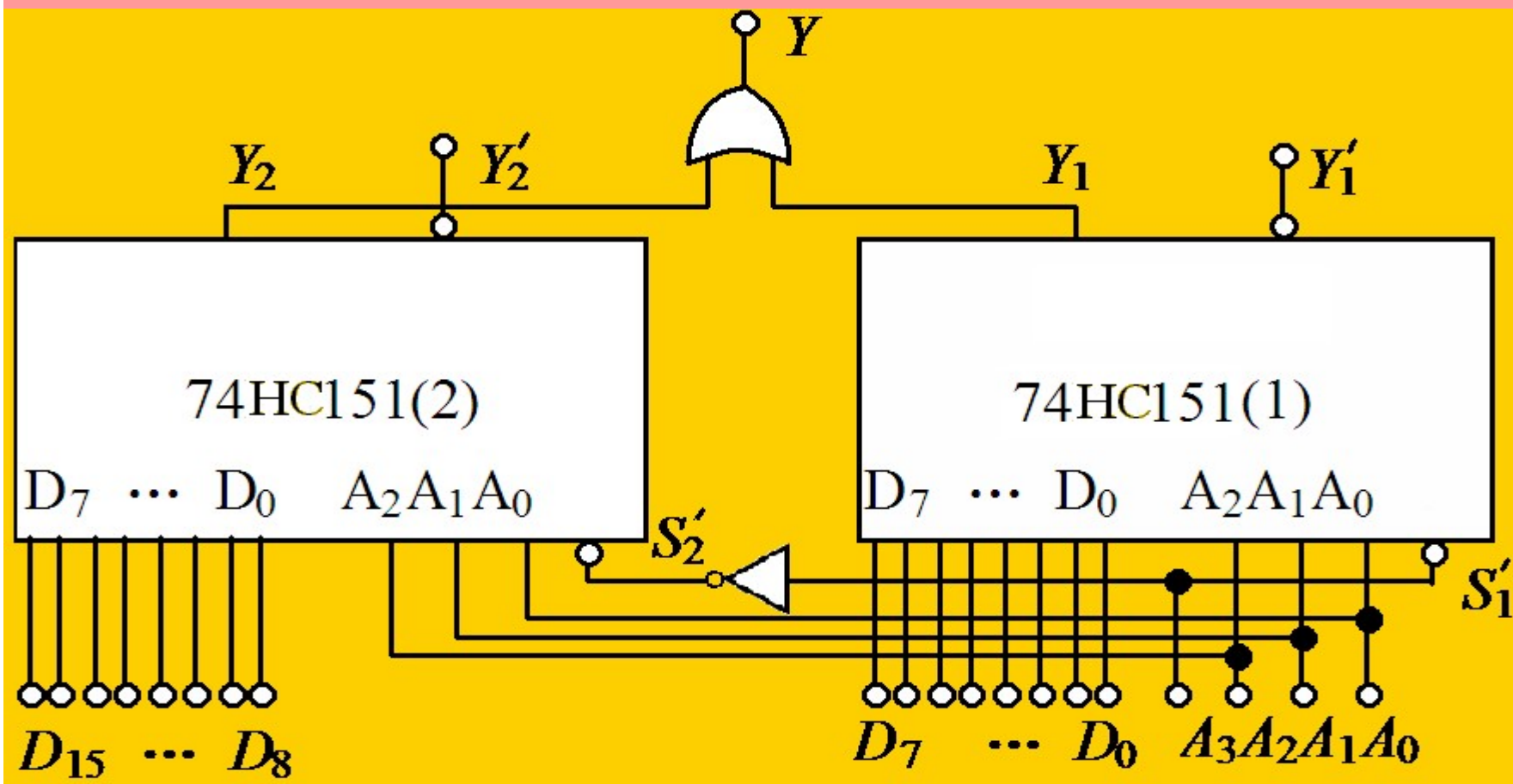
例4.3.4

用双4选1数据选择器构成8选1数据选择器。



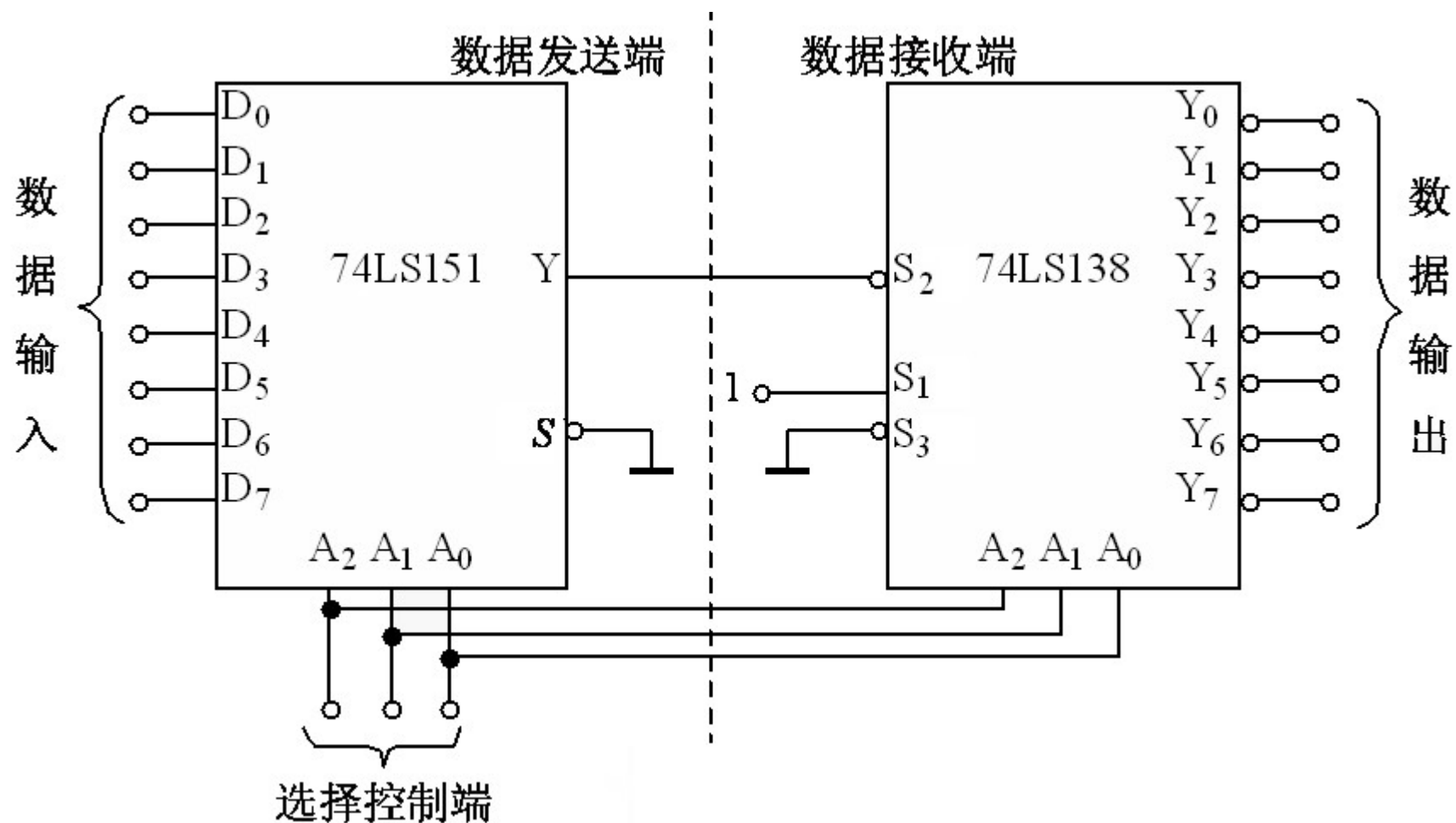
$A_2=1$ 时，下边一半数据选择器工作，数据 $D_4 \sim D_7$ 选择一路输出。

2片8选1数据选择器74LS151构成16选1的数据选择器



$A_3=0$ 时, $S'_1=0$ 、 $S'_2=1$, 片(2)禁止、片(1)工作
 $A_3=1$ 时, $S'_1=1$ 、 $S'_2=0$, 片(1)禁止、片(2)工作

数据分配器和数据选择器一起构成数据分时传送系统



§ 4.3.4 加法器

举例： $A=1101$, $B=1001$, 计算 $A+B$

$$\begin{array}{r} 1101 \\ + 1001 \\ \hline 10110 \end{array}$$



加法运算的基本规则：

- (1) 逢二进一。**
- (2) 最低位是两个数最低位的相加，不需考虑进位。**
- (3) 其余各位都是三个数相加，包括加数、被加数和低位来的进位。**
- (4) 任何位相加都产生两个结果：本位和、向高位的进位。**

1位加法器

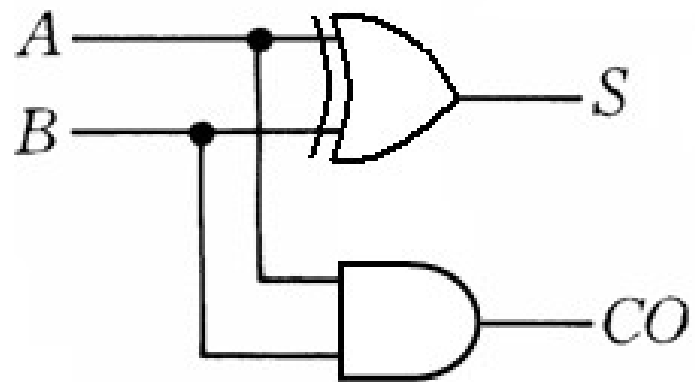
(1) 半加器: 半加运算不考虑从低位来的进位
 A ---加数; B ---被加数; S ---本位和; Co ---进位。

真值表

| A | B | Co | S |
|-----|-----|------|-----|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

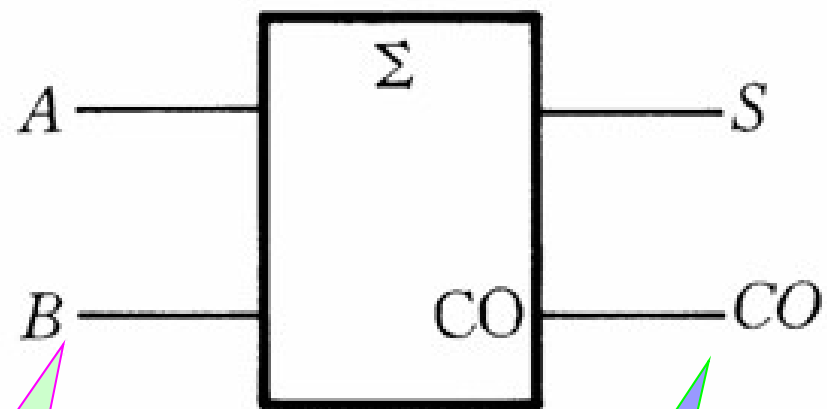
$$S = A'B + AB' = A \oplus B$$

$$Co = AB$$



(a)

2个输入端



(b)

2个输出
端

逻辑图

逻辑符号




(2) 全加器:

相加过程中, 既考虑加数、被加数又考虑低位的进位。

A ---加数; B ---被加数; C_i ---低位的进位;
 S ---本位和; C_0 ---进位。

逻辑状态表见下页

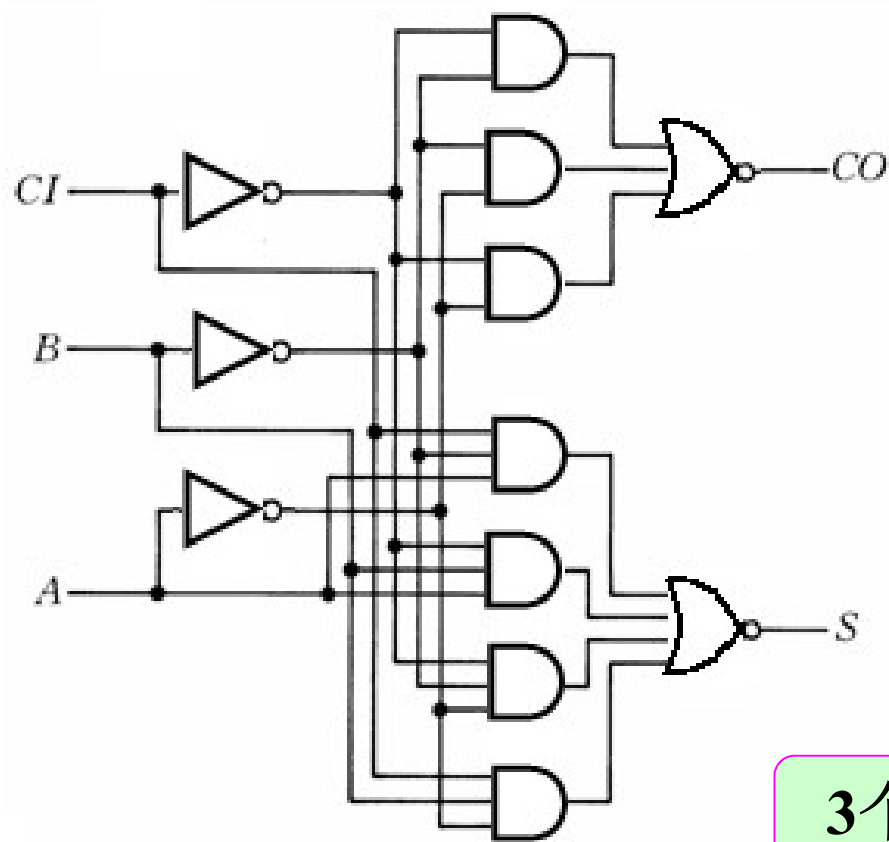


| C_i | A | B | S | C_o |
|-------|-----|-----|-----|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$$S = C_i' A' B + C_i' A B' + C_i A' B' + C_i A B = C_i \oplus A \oplus B$$

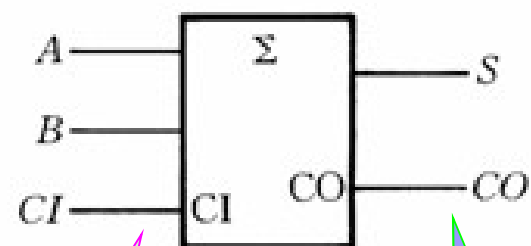
$$C_o = AB + C_i A + C_i B = AB + C_i (A \oplus B)$$

课本上采用了圈0的方法



(a)

双全加器74LS183



(b)

3个输入端

2个输出
端

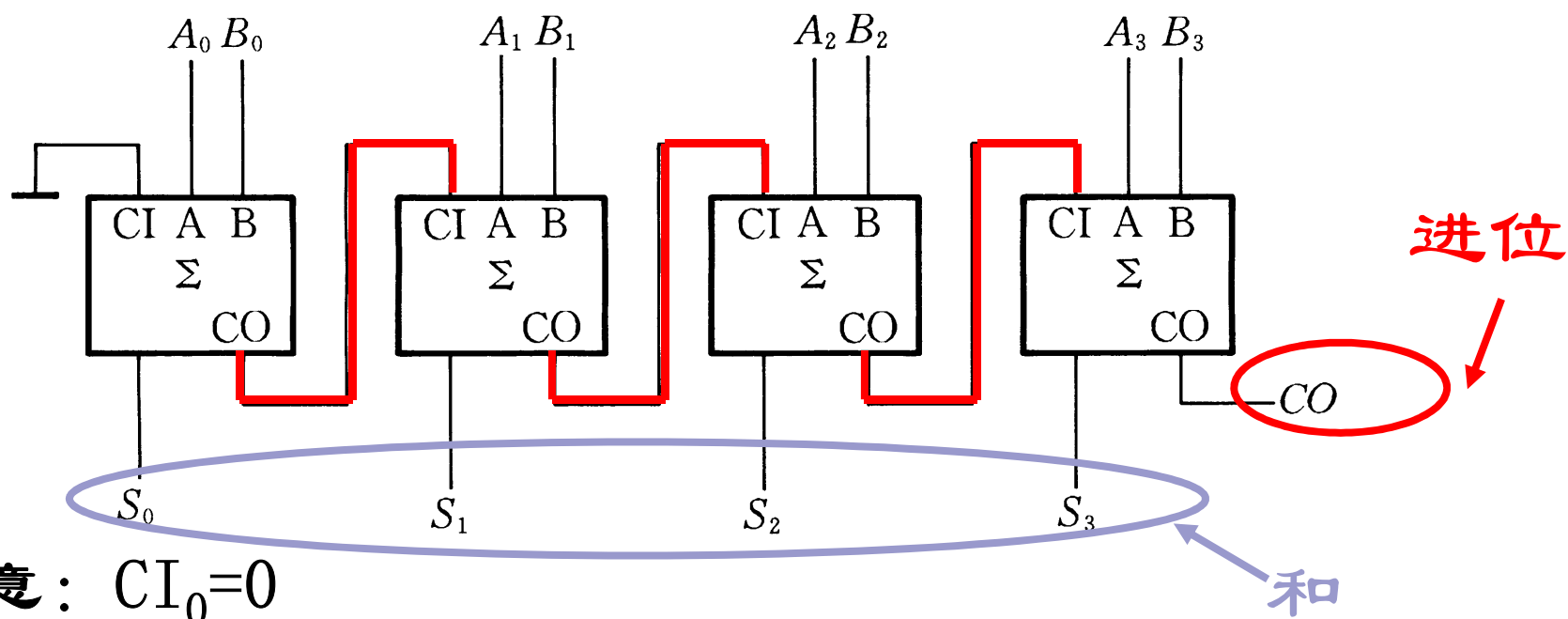
多位加法器

(1) 串行进位加法器

低位全加器进位输出

↓
高位全加器进位输入

如图：用全加器实现4位二进制数相加。



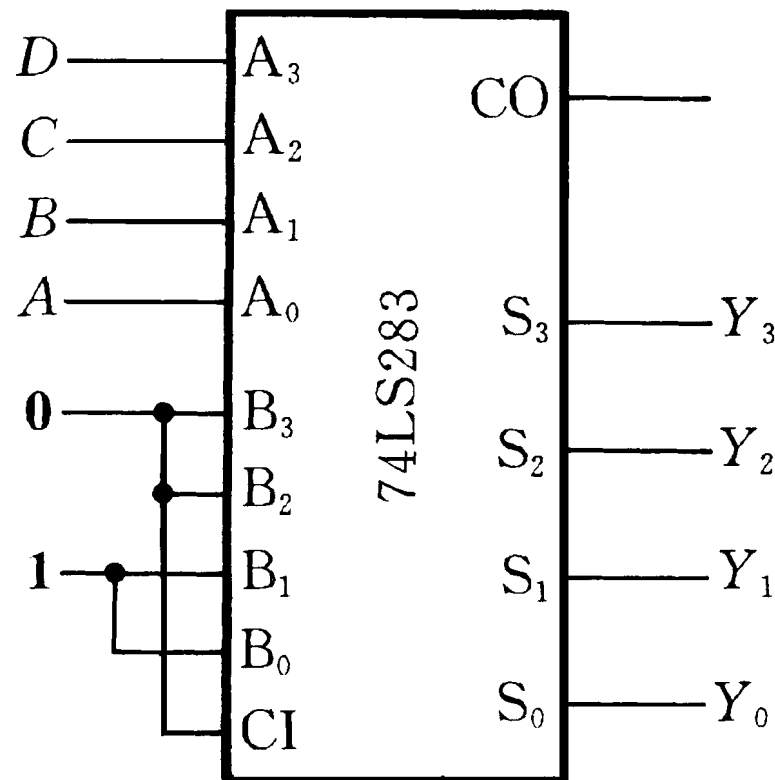
(2) 超前进位加法器

例4.3.7

解: **BCD码+0011=余3码**

设输入8421码用变量DCBA表示, 输出余三码用变量 $Y_3Y_2Y_1Y_0$ 表示。
则有

$$Y_3Y_2Y_1Y_0 = \text{DCBA} + 0011$$

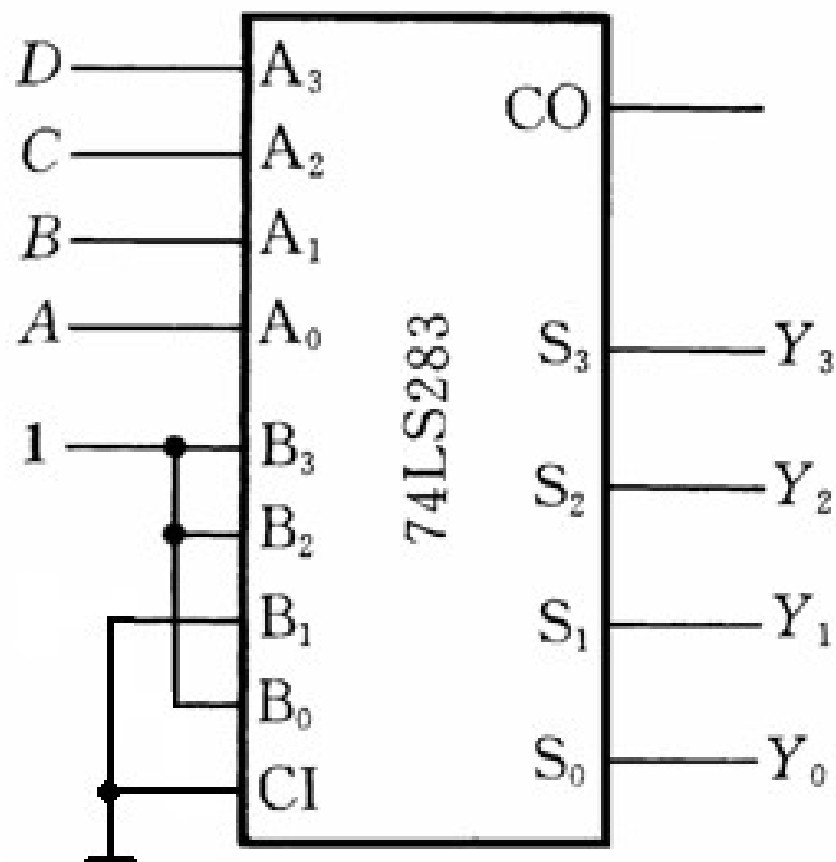


用一片74LS283将余三码转换成8421BCD码。

解： 余3码 - 0011 = BCD码

设输入余三码用变量 $DCBA$ 表示，输出8421码用变量 $Y_3Y_2Y_1Y_0$ 表示。则有

$$\begin{aligned} Y_3Y_2Y_1Y_0 &= DCBA + [0011]_{\text{补}} \\ &= DCBA + 1101 \end{aligned}$$



全减器真值表

***A*---被减数;**

***B*---减数;**

***C*---低位的借位;**

***D*---本位差;**

***J*---向高位的借位。**

| <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> | <i>J</i> |
|----------|----------|----------|----------|----------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



§ 4.3.5 数值比较器

定义：对两数**A**、**B**（可以是一位，也可能是多位）进行大小比较的逻辑电路。比较的结果有 **$A > B$** 、 **$A < B$** 、 **$A = B$** 三种结果。

一、1位数值比较器

设 $A > B$ 时 $Y_1 = 1$ ； $A < B$ 时 $Y_2 = 1$ ； $A = B$ 时 $Y_3 = 1$ 。

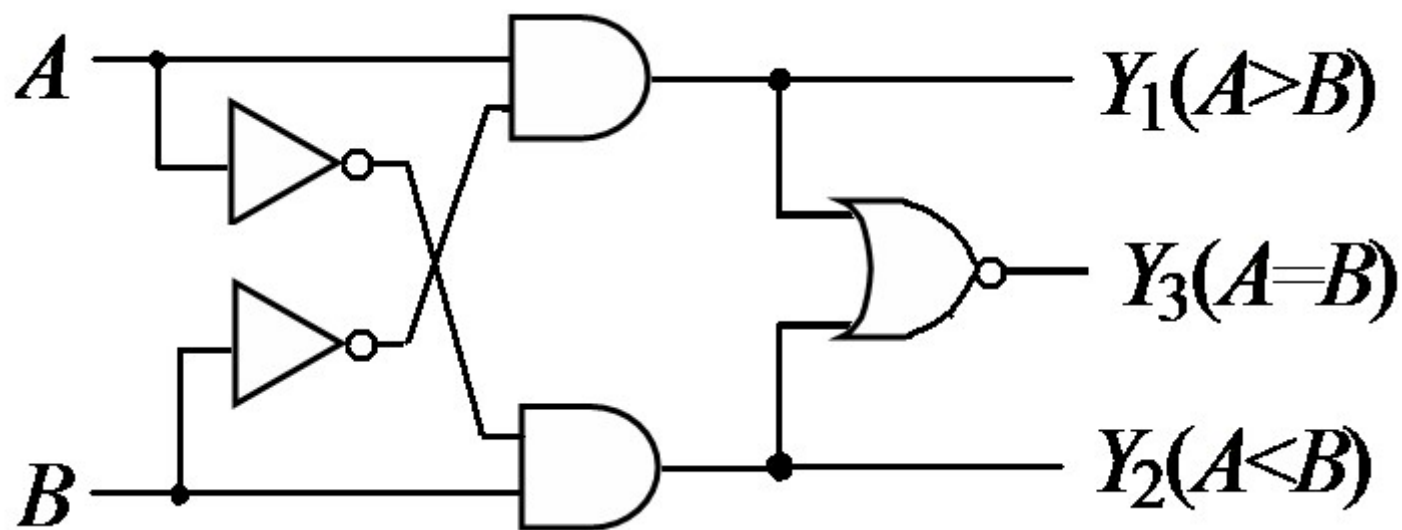
得1位数值比较器的真值表。

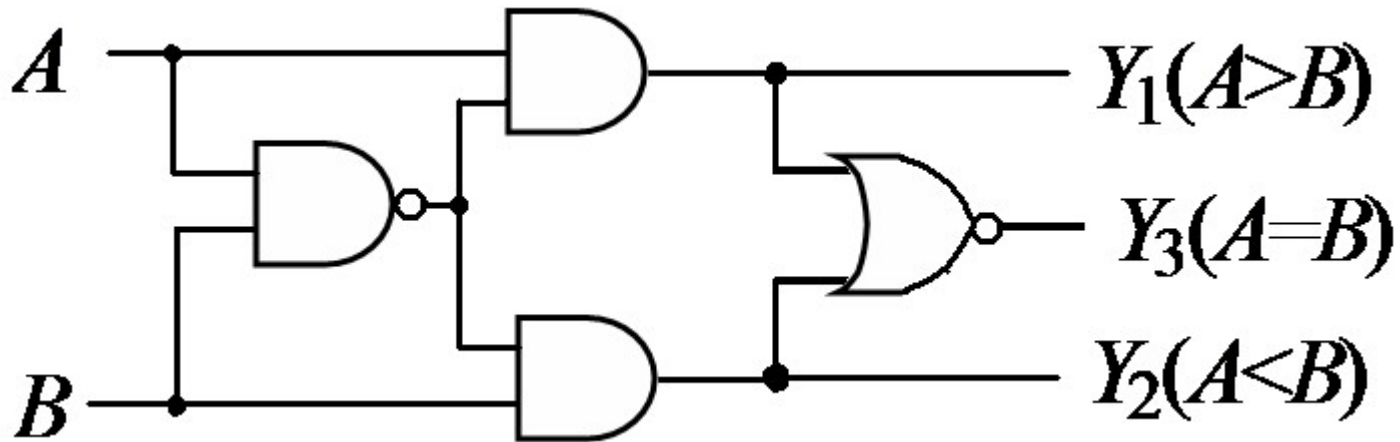
| A | B | $Y_1(A > B)$ | $Y_2(A < B)$ | $Y_3(A = B)$ |
|-----|-----|--------------|--------------|--------------|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |

逻辑表达式

$$\begin{cases} Y_1 = AB' \\ Y_2 = A'B \\ Y_3 = A'B' + AB = (A'B + AB')' \end{cases}$$

逻辑图





$$Y_1 = A \cdot (AB)' = A \cdot (A' + B') = AB'$$

$$Y_2 = B \cdot (AB)' = B \cdot (A' + B') = A'B$$

$$Y_3 = (AB' + A'B)' = AB + A'B'$$



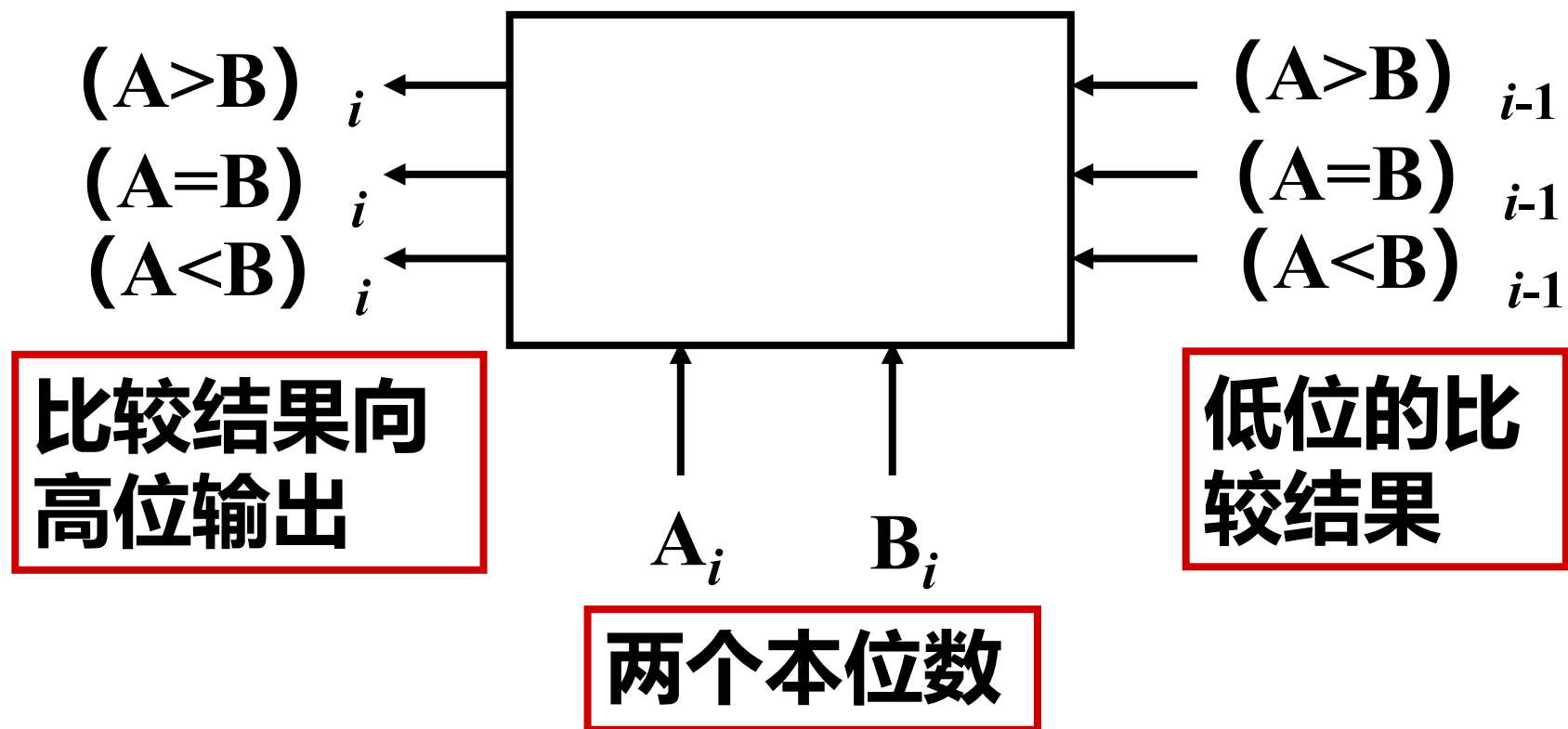
二、多位数值比较器

比较原则：

1. 先从高位比起,高位大的数值一定大。
2. 若高位相等,则再比较低位数,最终结果由低位的比较结果决定。

请根据这个原则设计一下：每位的比较应包括几个输入、输出？

A、B两个多位数的比较：



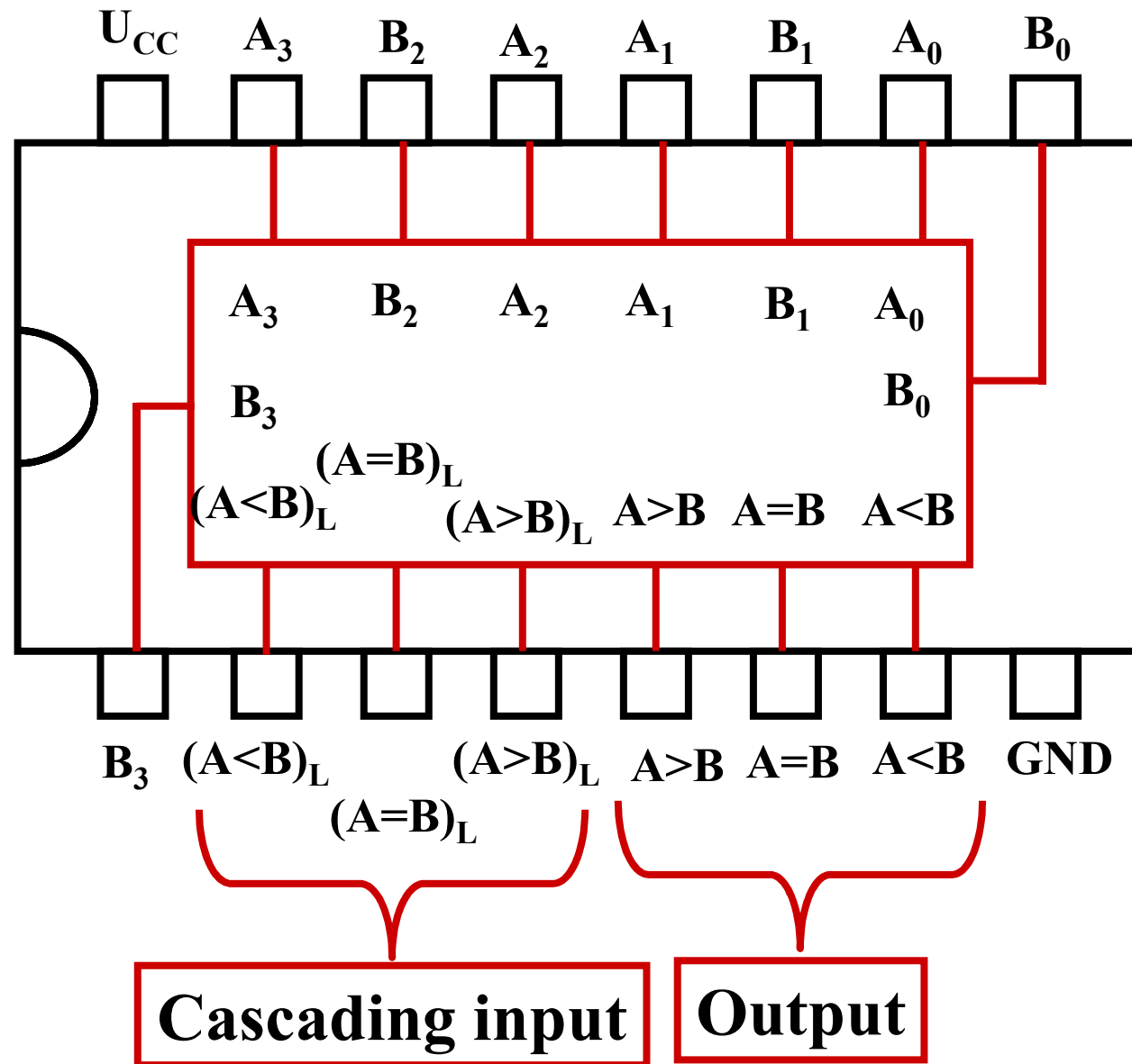
每个比较环节的功能表

| 输入 | | | | | 输出 | | |
|-----------|-------|--|---------------|---------------|-----------|-----------|-----------|
| A_i | B_i | $(A>B)_{i-1}$ | $(A=B)_{i-1}$ | $(A<B)_{i-1}$ | $(A>B)_i$ | $(A=B)_i$ | $(A<B)_i$ |
| 1 | 0 | ϕ | ϕ | ϕ | 1 | 0 | 0 |
| 0 | 1 | ϕ | ϕ | ϕ | 0 | 0 | 1 |
| $A_i=B_i$ | | 输出 $(A>B)_i$ 、 $(A=B)_i$ 和 $(A<B)_i$ 分别等于 $(A>B)_{i-1}$ 、 $(A=B)_{i-1}$ 和 $(A<B)_{i-1}$ | | | | | |

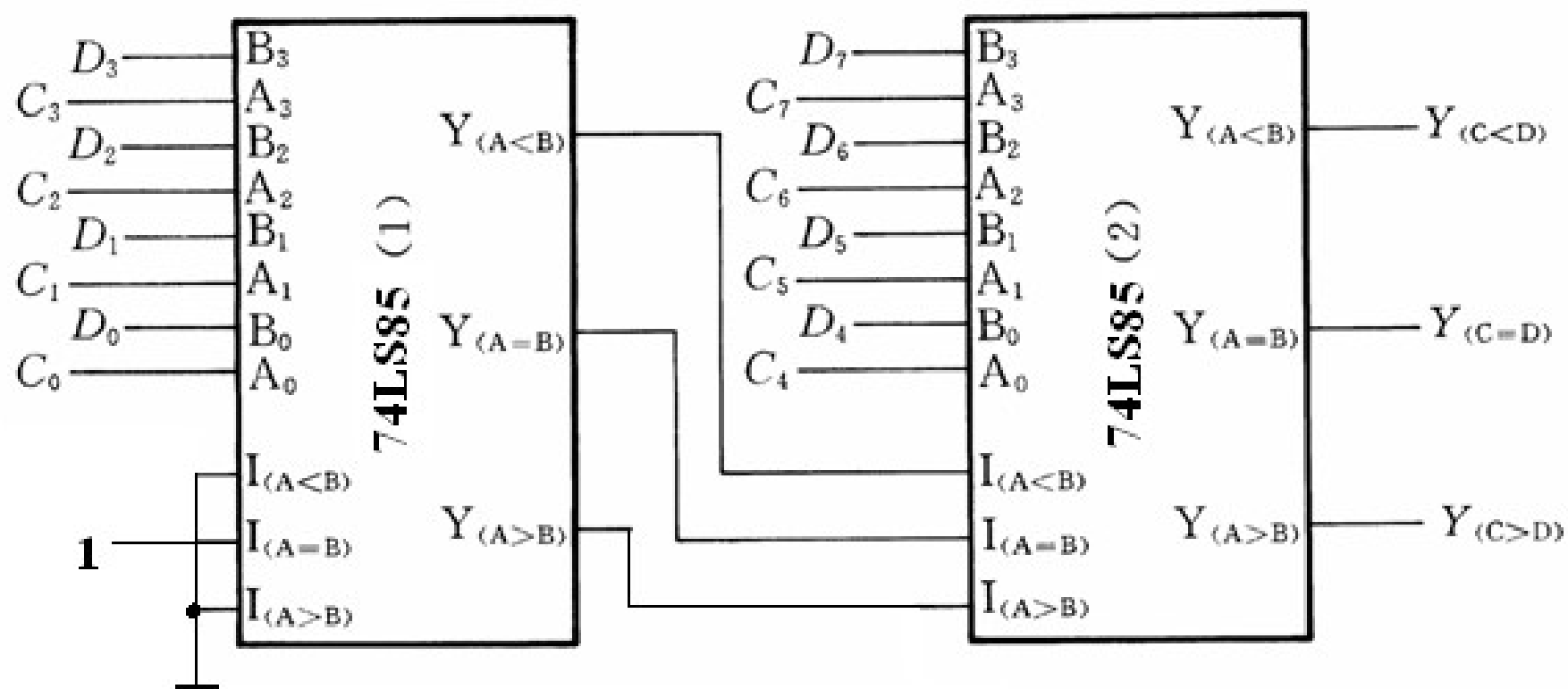
四位集成电路比较器74LS85

| Comparing Inputs | | | | Cascading Inputs | | | Outputs | | |
|---------------------|---------|---------|---------|---------------------|-------|-------|---------|-------|-------|
| A3, B3 | A2, B2 | A1, B1 | A0, B0 | A > B | A < B | A = B | A > B | A < B | A = B |
| A3 > B3 | X | X | X | X | X | X | H | L | L |
| A3 < B3 | X | X | X | X | X | X | L | H | L |
| A3 = B3 | A2 > B2 | X | X | X | X | X | H | L | L |
| A3 = B3 | A2 < B2 | X | X | X | X | X | L | H | L |
| A3 = B3 | A2 = B2 | A1 > B1 | X | X | X | X | H | L | L |
| A3 = B3 | A2 = B2 | A1 < B1 | X | X | X | X | L | H | L |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 > B0 | X | X | X | H | L | L |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 < B0 | X | X | X | L | H | L |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | H | L | L | H | L | L |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | L | H | L | L | H | L |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | L | L | H | L | L | H |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | X | X | H | L | L | H |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | H | H | L | L | L | L |
| A3 = B3 | A2 = B2 | A1 = B1 | A0 = B0 | L | L | L | H | H | L |

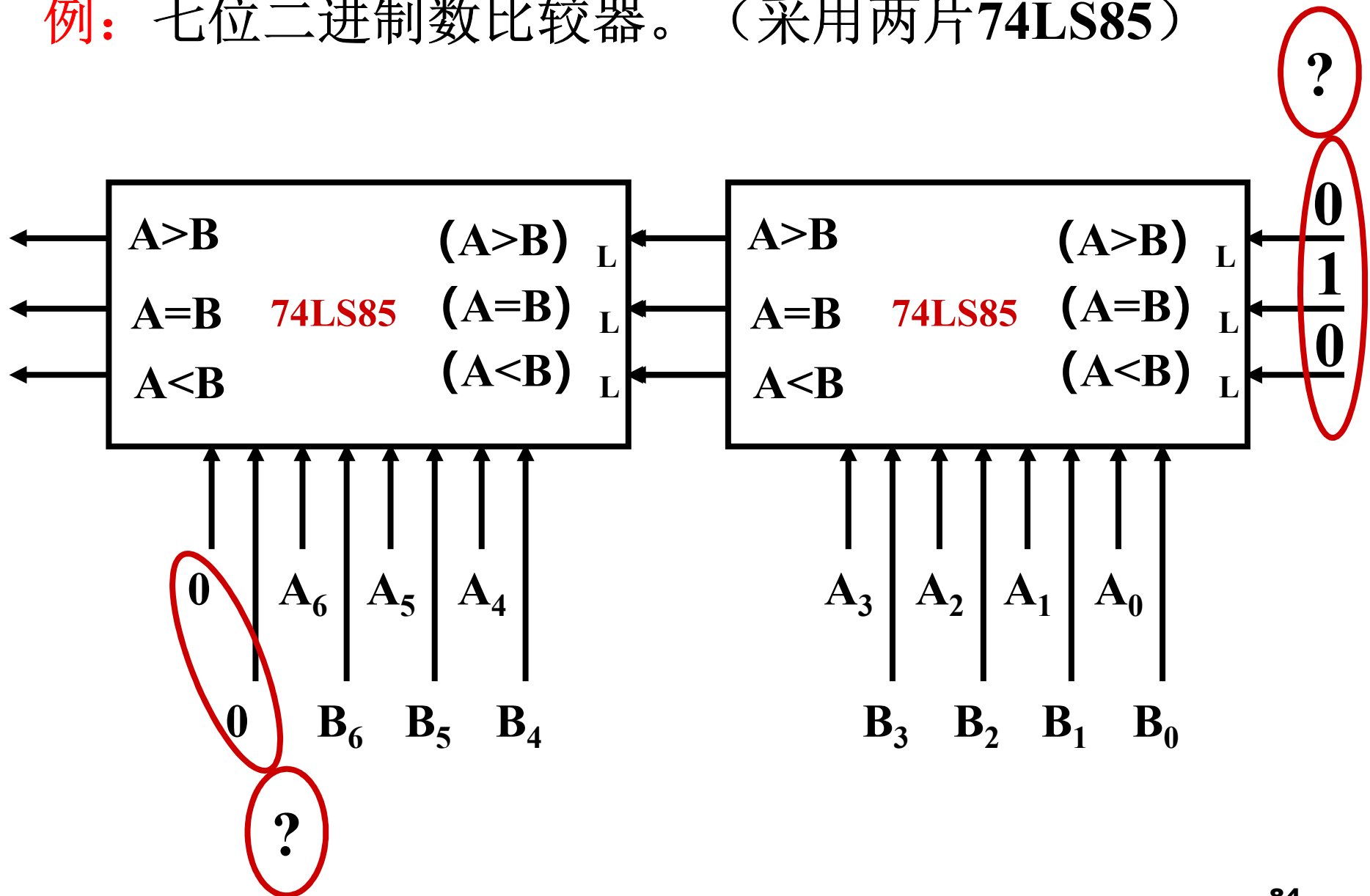
四位集成电路比较器74LS85




例 试用两片74LS85组成一个8位数值比较器。



例：七位二进制数比较器。（采用两片74LS85）





例：设计三个四位数的比较器，可以对A、B、C进行比较，能判断：

- (1) 三个数是否相等。
- (2) 若不相等，A数是最大还是最小。

比较原则：

先将A与B比较，然后A与C比较，若A=B
A=C，则A=B=C；若A>B A>C，则A最大；若
A<B A<C，则A最小。

可以用两片74LS85实现。

